4-19-2013

Multiple-target tracking using spectropolarimetric imagery

Tingfang Zhang

Multiple-Target Tracking Using Spectropolarimetric Imagery

by

Tingfang Zhang

A thesis submitted in partial fulfillment of the requirements

for the degree of Master of Science

in the Chester F. Carlson Center for Imaging Science,

College of Science,

Rochester Institute of Technology

19 April, 2013

Signature of the Author_____

Accepted by_____

Director, M.S. Degree Program Date

CHESTER F.CARLSON
CENTER FOR IMAGING SCIENCE
COLLEGE OF SCIENCE
ROCHESTER INSITITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

M.S DEGREE THESIS

The M.S. degree Thesis of Tingfang Zhang
has been examined and approved by the
thesis committee as satisfactory for the
thesis requirement for the
Master of Science degree

Dr. John Kerekes, Thesis Advisor

Dr. Anthony Vodacek, Committee Member

Dr. Zoran Ninkov, Committee Member

Date

Multiple-Target Tracking Using Spectropolarimetric Imagery

by
Tingfang Zhang

Submitted to the
Chester F. Carlson
Center for Imaging Science
College of Science
in partial fulfillment of the requirements
for the Master of Science Degree
at the Rochester Institute of Technology

ABSTRACT

Detection and tracking methods are two hot research topics in the field of multiple target tracking. Often change detection and motion tracking are used to detect and track moving vehicles, but in this thesis new approaches are provided to improve these two aspects. In the detection aspect, a combined detection method is presented to improve target detection techniques. The method of combining RX (Reed-Xiaoli*)* with change detection has demonstrated good performance in highly cluttered, dynamic ground-based scenes. In the tracking aspect, Kalman filter and Global Nearest Neighbor are applied in motion tracking to predict the location and implement data association respectively. Spectral features are extracted for each vehicle to solve the limitation of motion tracking through feature matching. The Bhattacharyya distance is used as a criterion in the feature matching procedure.

Our algorithm has been tested using three sets data. One is a set of multispectral polarimetric imagery acquired by the Multispectral Aerial Passive Polarimeter System (MAPPS). Another two data sets are spectropolarimetric imagery generated by the Digital Imaging and Remote Sensing Image Generation tool. The tracking performance is analyzed by calculating performance metrics: track purity and (Multiple Object Tracking Accuracy) MOTA. For MAPPS data, the average MOTA and track purity of feature-aided tracking increase 1 percent and 9 percent over those of motion-only tracking respectively. For DIRSIG data with trees, the average track purity of feature-aided tracking in without noise case increases 2 percent over that of motion-only tracking. In this work, we have demonstrated the capability of detection and tracking methods applied in a complex environment.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

1.1 Background

Multiple moving target tracking of image sequences is one of the most important subjects in computer vision. It has been applied in many computer vision fields, such as military guidance, safety detection, video surveillance, and so on. The challenge problems for multiple target tracking are to solve the complex conditions, such as the inter-object occlusion, partial occlusion of the targets by background obstacles, splits and loss of tracking. Traditional tracking methods (Fenton, et al., 1976; Cormier, et al., 1980; Shin, 1990) are mainly kinematic, depending on the location and velocity, and fail to track the targets correctly in many situations.

In recent years, various tracking methods have been proposed in the literature to solve these cases. Nguyen et al. (2003) and Chang et al. (2001) use multiple cameras to obtain continuous visual information of targets so that they can be tracked through interactions. Medioni et al. (2001) proposed an approach based on graph theory for tracking multiple targets. But this method only considers split tracks. Other researches proposed to use geometric features to solve occlusion, merge or split. Li X. et al. (2010) regards the center of mass and the area of targets as the feature to track the targets. However, not all of these methods can solve all these cases.

Recent advance in electronics and sensor design have enabled the capture of multispectral, hyperspectral, polarization and spectropolarimetric imagery. Multispectral and hyperspectral sensors provide multiple or high spectral resolution imagery, which enhance their ability to identify objects from the background. Therefore, spectral information has been used as the feature in the literature in recent years as a way to solve problems with a kinematic-only tracker. Blackburn et al. (2007) uses hyperspectral feature aided tracking that demonstrates better performance than a traditional kinematic tracker. Varsano et al (2010) similarly use

hyperspectral images for point-target tracking. Polarization information obtained from the surface reflections of the target yields unique and discriminatory signatures which can augment the target detection. Harchanko at el. (2005) use imaging polarimetry to detect water-surface objects. EI-Saba at el. (2004) improves target detection using polarization enhancement. A partnership between Numerica Corporation and the RIT Center for Imaging Science (CIS) is exploiting multi-modality imagery, including panchromatic, hyperspectal and polarimetric, to track vehicles in a highly cluttered, dynamic scenes. This work is in response to the U.S. Air Force Office of Scientific Research (AFOSR) Discovery Challenge Thrust (DCT) on performance-driven sensing, which is described in the next section.

1.2 Performance-Driven Sensing

AFOSR is the sponsor for a Discovery Challenge Thrust (DCT) in the field of integration of adaptive hardware with the modeling of target scene phenomenology, data processing and exploitation algorithms (Reinhardt, 2007). AFOSR is interested in research in a performance-driven sensing context in order to solve problems of detecting and tracking targets in a complex environment. The concept of performance-driven sensing is to minimize unnecessary sensor use by intelligently selecting an optimum subset of sensors that are most relevant for target tracking.

There are three research veins (Kerekes et al., 2007) taken by the RIT-Numerica research team. First, modeling of scene phenomenology and moving targets is regarded as a simulated input for tracking. Second, a focus on the research in the device and optical system is pursued in order to achieve multi-modality imagery. Third, a performance-driven target tracking algorithm is applied to track moving targets in a scene. Several papers summarizing this work have been published (Kerekes et al., 2009; Presnar et al., 2010). This thesis continues the above research

and focuses on the algorithm research in the third vein to track the vehicles using spectropolarimetric imagery.

1.3 Dissertation Proposal Overview

Chapter 2 briefly describes the objective of this thesis. Chapter 3 introduces the data sets used for tracking. Chapter 4 describes the detection and tracking algorithms. Chapter 5 discusses the resultant imagery and metrics of the tracking. Chapter 6 draws the conclusion and Chapter 7 proposes the future work.

2 Objectives

Conventional tracking methods are mostly kinematic relying on measurements of the location and velocity of the vehicle. Unsatisfying tracking results are often produced due to complex conditions, such as inter-object occlusion, partial occlusion of the objects by background obstacles, splits or missed detection. Some methods have been proposed to solve the problems of kinematic tracking by combining features with motion tracking. But the features used are usually shape (Li et al., 2010) or gradient direction (Lichtenauer et al., 2005), which are not robust in the process of tracking because they vary between frames. Other features, such as polarization and spectral information, have been brought forward to tracking vehicles in some academic papers in the recent years (Rice et al., 2009), and have demonstrated good performance.

As the reflection or emission of the objects within the scene, the polarization state of radiation is sensitive to the surface features such as the smoothness or roughness, shapes, edges and so on (EI-Saba at el., 2010). Therefore, polarization information obtained from the surface reflections of the target has the potential to enhance the target detection.

Multispectral imaging (MSI) provides both spectral and spatial information about the target, which are very useful for target detection and tracking. In addition, polarization spectral imaging (PSI), also known as spectropolarimetry imaging, containing both the spectral and polarization information, can be effectively exploited to highlight both contributions for more efficient target detection and tracking.

Therefore, in this thesis, both spectral and polarization information of spectropolarimetic data are applied to solve the problems resulting from motion tracking. The specific objectives of this research are:

1. Study the image registration method done in the spatial domain.
2. Study the detection methods, such as RX detection and change detection algorithms, to detect vehicles.
3. Demonstrate how polarization information can improve the detection results.
4. Investigate the Global Nearest Neighbor data association method.
5. Demonstrate how the feature-aid tracking method can solve the problems of motion-only tracking.
6. Investigate how sensor noise affects the tracking results.

3 Background and Data Analysis

In this thesis, three data sets are exploited to test the tracking algorithm. Two data sets, with trees and without trees respectively, are generated by the Digital Imaging and Remote Sensing Image Generation (DIRSIG) software package (Schott et al., 1999). Another data set was collected by the Multispectral Aerial Passive Polarimeter System (MAPPS) (Bartlett et al., 2011). The MAPPS data set is raw data. As the air plane moves forward, a filter wheel spins, collecting frames of imagery in various wavebands and polarized states. Thus adjacent frames have a translation and small rotation, and image registration has to be implemented on this data set before using it for tracking. In this chapter, background on polarization and multispectral imaging, as well as the two data sets and image registration method, are introduced.

3.1 Polarization

The polarization properties of electromagnetic radiation can be completely described by the Stokes vector that consists of four components known as the Stokes parameters:

$$S = [S_0 \quad S_1 \quad S_2 \quad S_3]^T \tag{3-1}$$

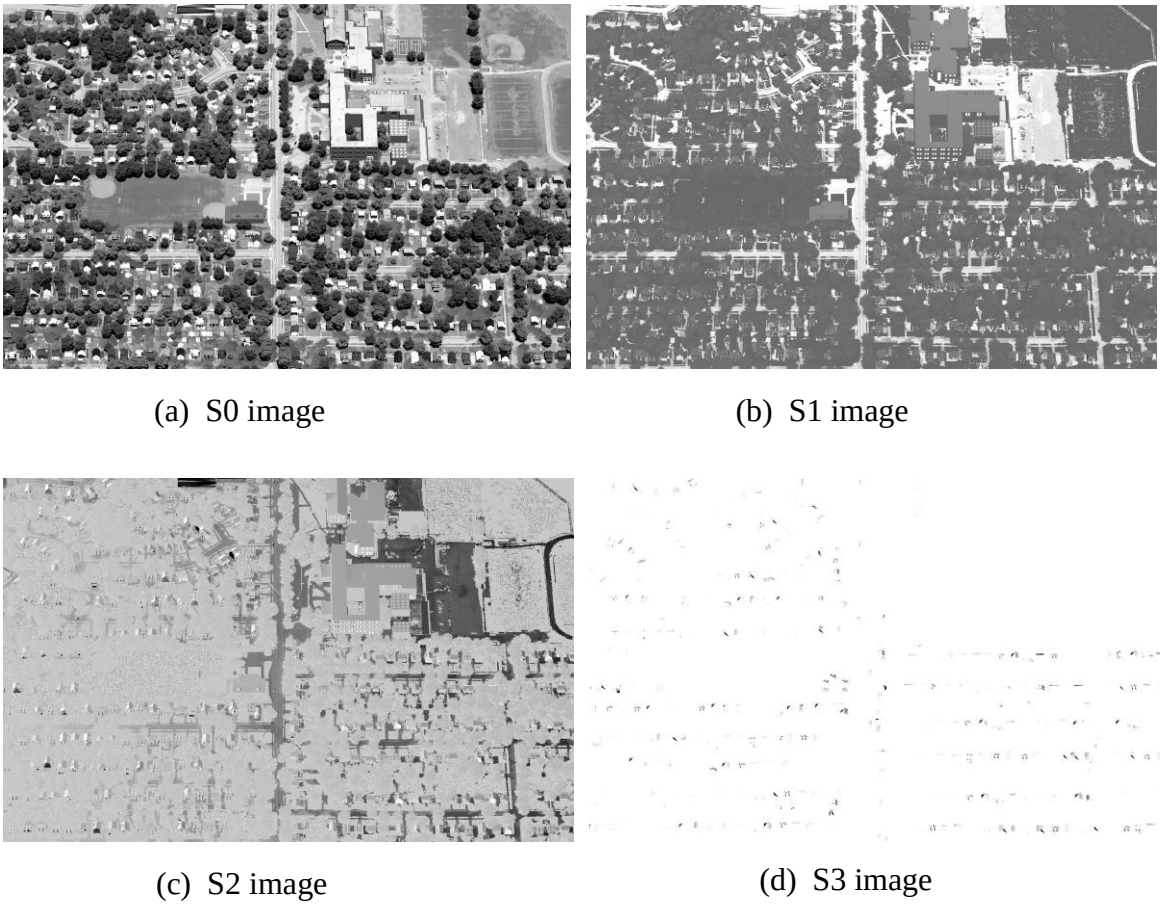
S_0 is the total intensity image.

$$S_0^2 = S_1^2 + S_2^2 + S_3^2 \tag{3-2}$$

S_1, S_2 and S_3 images describe the linear polarization, orientation and circular polarization content of light in the scene respectively. These Stokes images can be calculated using the intensity images.

$$S = \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} = \begin{bmatrix} I(0^o) + I(90^o) \\ I(0^o) - I(90^o) \\ I(45^o) - I(135^o) \\ I(Rcirc) - I(Lcirc) \end{bmatrix} \tag{3-3}$$

where I(0^o), I(45^o), I(90^o), and I(135^o) are the measured intensity images acquired with a corresponding polarizer. I(Rcirc) and I(Lcirc) are the right-circular and left-circular polarization respectively. S_1 represents the difference of intensity between horizontal and vertical linearly polarized components. S_2 is the difference between 45^o and 135^o linearly polarized components. S_3 is the difference between the right circular and the left circular component. The Stokes parameters are related by equation 3-2. Only three of them are independent. Fig. 3-1 shows the four Stokes images of the green band of the second frame of DIRSIG data.



(a) S0 image (b) S1 image

(c) S2 image (d) S3 image

Figure 3-1: Four Stokes parameter images of the second frame of DIRSIG data

Based on the Stokes parameters, other geometric features have been derived and used to characterize and explore the polarization phenomenon, such as the angle of polarization and the

degree of polarization. The angle of polarization is calculated as in the equation below, depending on S_2 and S_1.

$$AoP = \frac{1}{2}\tan^{-1}(\frac{S_2}{S_1}) \tag{3-4}$$

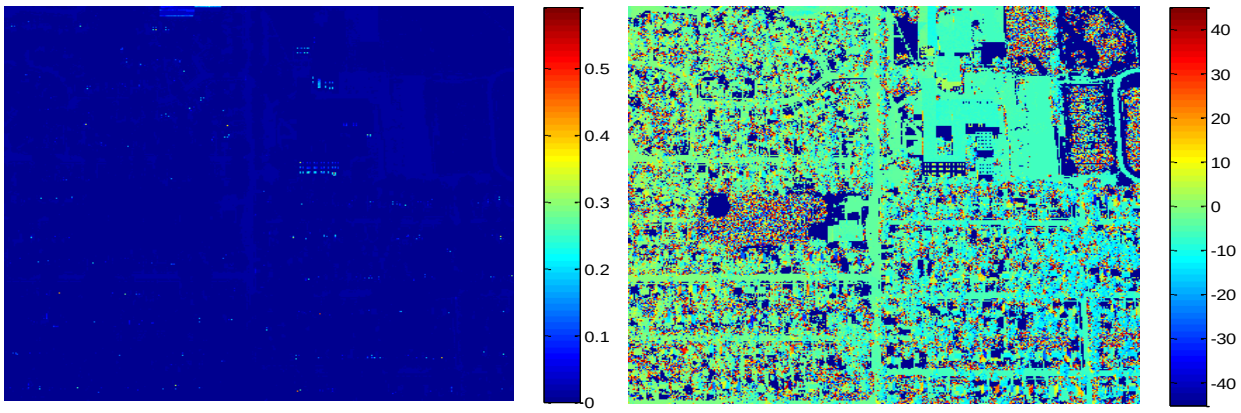
The degree of polarization is calculated as:

$$DoP = \frac{\sqrt{S_1^2 + S_2^2 + S_3^2}}{S_0} \tag{3-5}$$

where DoP is between 0 and 1. According to the value of DoP, three conditions are classified: randomly polarized (DoP= 0), completely polarized (DoP= 1) and partially polarized (0<DoP<1). Compared with the other three Stokes parameters, S_3 is typically small. So when performing remote sensing of the Earth, S_3 can be neglected. We can simplify equation 3-5 to Degree of Linear Polarization (DoLP):

$$DoP \approx DoLP = \frac{\sqrt{S_1^2 + S_2^2}}{S_0} \tag{3-6}$$

Fig. 3-2 show the DoP and AoP images respectively, calculated using the Stokes parameter images shown in Fig. 3-1.



(a) Degree of Polarization image (b) Angle of Polarization image

Figure 3-2: Degree of Polarization image and Angle of Polarization image

3.2 Multispectral Imaging

Multispectral imaging is a three-dimensional data cube, which is generated using several spectral vectors for each pixel. The spectral vectors range from the visible region (0.4-0.7μm) to the infrared (0.7-1μm). Each channel of the data cube can be displayed as a grayscale image and the combinations of two or three channels as a color image. If the blue, green and red channels are extracted, a true-color picture can be displayed.

Multispectral imaging contains both spatial and spectral information. If all the pixels of a single ground resolution cell are extracted and one plots the spectral values as a function of wavelength, the spectral information for that ground resolution cell is obtained. If all the pixels in the same spectral band are extracted, the intensity values show the spatial distribution of reflectance of the scene for that particular wavelength (EI-Sabe at al., 2010). Multispectral imaging has been widely used in the remote sensing field for detection and tracking.

3.3 DIRSIG Data Set

DIRSIG, a physics-based synthetic image simulation software package, has been developed by RIT CIS researchers and scientists. It has the ability to produce imagery in a variety of modalities, such as multi-spectral, hyper-spectral and spectropolarimetric imagery, which can be used to test the performance of spatial and spectral image analysis algorithms. DIRSIG contains a graphical user interface editor which has five major simulation components and additional options (Brown, 2006). They are scene modeling, atmosphere condition modeling, imaging platform modeling, platform motion modeling, data collection and options respectively, as shown in Fig. 3-3. The details about these simulation components are not introduced in this thesis. More details about DIRSIG can be found in the DIRSIG manual (Brown, 2006).

Figure 3-3: Five major simulation components plus Options in DIRSIG GUI
(From Presnar, 2010)

The DIRSIG data set is all nadir imagery in this project, and are formed by assuming a static sensor mounted on an aerial platform pointing directly toward a target stare point. Pixel pitches are simulated as Δx=Δy=17 [μm/pix] square for the 2640*1680 image array. The flying height H is equal to 2417 [m] AGL and the focal length is 204 [mm]. Based on the focal length and flying height, the scale factor (s) can be calculated:

$$s = \frac{f}{H} = \frac{204 \times 10^{-3}[m]}{2417[m]} = 8.4 \times 10^{-5} \tag{3-7}$$

The ground sample distance (GSD) can be computed:

$$GSD_X = GSD_Y = \frac{\Delta x}{s} = \frac{17 \times 10^{-6}[m]}{8.4 \times 10^{-5}} = 0.20[m] \tag{3-8}$$

The detector array physical dimensions for an 2640*1680 pixel detector:

$$l_x = n_x \times \Delta x = 2640\,[pix] \times 17[\mu m / pix] = 44.88\,[mm] \tag{3-9}$$

$$l_y = n_y \times \Delta y = 1680\,[pix] \times 17[\mu m / pix] = 28.56\,[mm] \tag{3-10}$$

The total ground scene dimensions:

$$l_X = \frac{l_x}{s} = \frac{44.88\,[mm]}{8.4 \times 10^{-5}} = 534\,[m] \tag{3-11}$$

$$l_Y = \frac{l_y}{s} = \frac{28.56\,[mm]}{8.4\times10^{-5}} = 340\,[m] \tag{3-12}$$

The total field of view of the sensor is computed using the trigonometric relationship between the focal length and the detector array dimensions.

$$FOV_x = 2\times HFOV_x = 2\times\tan^{-1}(\frac{l_x}{2f}) = 2\times\tan^{-1}(\frac{44.88\,[mm]}{2\times204\,[mm]}) = 2\times6.277^o = 12.56^o \tag{3-13}$$

$$FOV_y = 2\times HFOV_y = 2\times\tan^{-1}(\frac{l_y}{2f}) = 2\times\tan^{-1}(\frac{28.56\,[mm]}{2\times204[mm]}) = 2\times4.004^o = 8^o \tag{3-14}$$

Each frame of this data set has 6 bands. The spectral range is 0.45 to 0.95 [um], with an interval of 0.1 um. Further, polarized imagery includes four Stokes parameters corresponding to each spectral band as simulated as well. Therefore, one frame of data has a total of 24 high resolution images. There are 30 sequential frames data for this project to track vehicles. The RGB bands (9th, 5th, 1st) of each frame can be extracted and adjusted for display together, as shown below in Fig. 3-4.

Figure 3-4: RGB image for the second frame of DIRSIG data

Different vehicles were simulated in DIRSIG with different vehicle paint. There are a total of 21 vehicles in this project. Two of them are shown in the figures below.

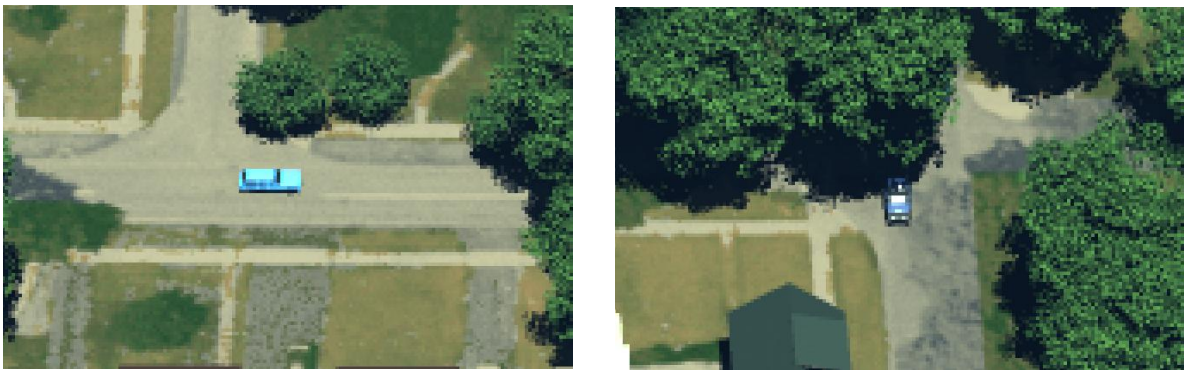


Figure 3-5: RGB image of vehicles simulated in the DIRSIG images used for performance evaluation

There is no noise in the generated DIRSIG data sets. To see how the noise affects the detection and tracking methods, a zero-mean Gaussian noise is added into data. The variance of Gaussian noise is calculated using the following equation:

$$\sigma^2 = (\frac{\mu}{SNR})^2 \tag{3-15}$$

where μ is the mean of intensity value of image, SNR is signal to noise ratio. Suppose SNR is equal to 30 for the DIRSIG data with trees, the mean is calculated for each image in a frame. Then using equation 3-15, the variance of each image is calculated. Final, Gaussian noise is added into each image.

3.4 MAPPS Data Set

The MAPPS instrument can produce multi-spectral polarimetric imagery of scenes for use in algorithm development and phenomenology studies. The main components of MAPPS consist of a digital camera, two filter wheels, system enclosure and so on, as shown in Fig. 3-6. The first filter wheel contains a set of linear polarizers, each oriented with their transmission axes rotated relative to each other. The second filter wheel contains a set of spectral bandpass filters and can

allow for up to 10 different spectral bands to be collected. As each spectral filter and the linear polarizer is rotated, spectral Stokes imagery can be collected (Bartlett et al., 2011).

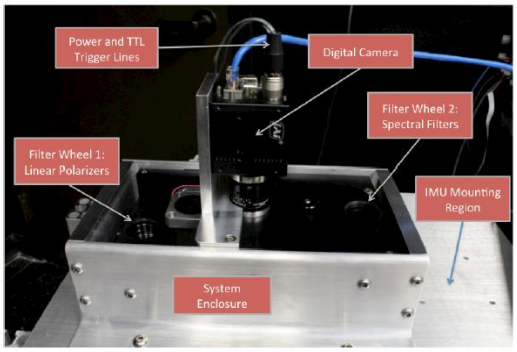


Figure 3-6: Illustration of the MAPPS system (From Bartlett et al., 2011)

When collecting the data set, the airplane flies at 4000 meters, the focal length of the digital camera is 35mm and the pixel pitch is 3.45 um. According to Equation 3-8 and 3-7, the scale factor *s* and the ground sample distance can be calculated respectively:

$$s = \frac{f}{H} = \frac{35\times10^{-3}[m]}{4000[m]} = 8.75\times10^{-6} \tag{3-16}$$

$$GSD_X = \Delta X = GSD_Y = \Delta X = \frac{\Delta x}{s} = \frac{3.45\times10^{-6}[m]}{8.75\times10^{-6}} = 0.394[m] \tag{3-17}$$

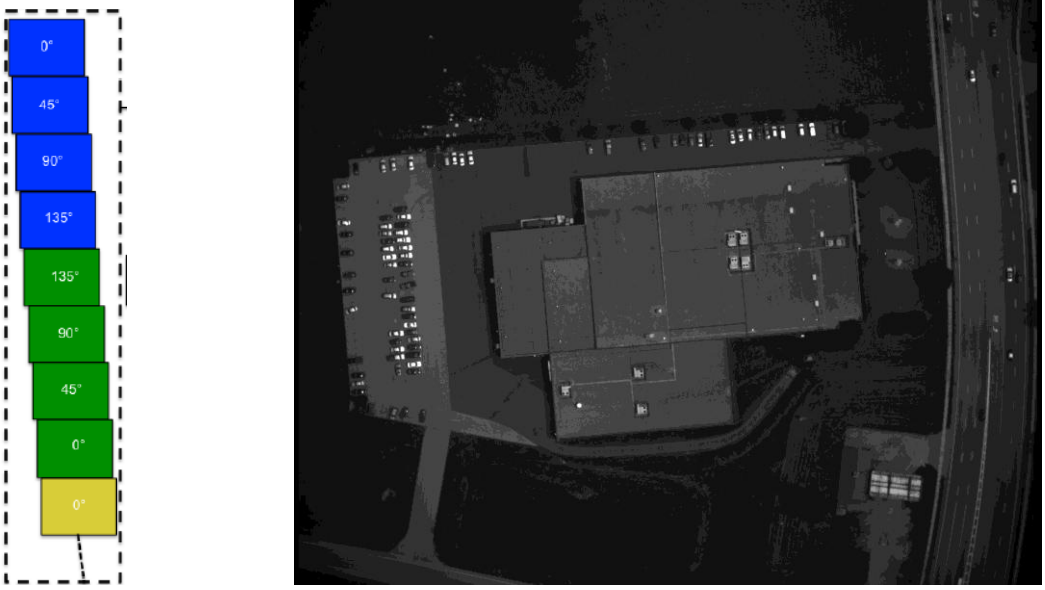
The detector array physical dimensions for an 2456*2058 pixel detector:

$$l_x = n_x \times \Delta x = 2456\,[pix] \times 3.45[\mu m / pix] = 8.47\,[mm] \tag{3-18}$$

$$l_y = n_y \times \Delta y = 2058\,[pix] \times 3.45[\mu m / pix] = 7.1\,[mm] \tag{3-19}$$

Only six spectral bands are chosen among the ten spectral filter set: blue (475nm), green (550nm), yellow (607nm), red (655nm), rededge (716nm) and NIR2 (809nm). The polarization filter set consists of four linear polarizers which allow generating four polarization images corresponding to the angle 0^o, 45^o, 90^o, and 135^o respectively. This data set has 24 frames since

each spectral band consists of four polarizer images. Each frame is a one band one polarization high resolution panchromatic image with dimensions of 2456*2058. Since each frame is obtained from the sensor at a different time as the plane moves forward, image registration is necessary to obtain accurate registered imagery. Fig. 3-7 (a) illustrates the sequence of MAPPS data and (b) is the first frame image (blue band at 0^o polarization).



(a) Image Sequence (From Bartlett et al., 2011) (b) The first frame Image

Figure 3-7: Illustration of sequence images MAPPS system and the first frame image

3.5 Image Registration

Image registration is the process of overlaying images of the same scene taken at different times, from different viewpoints, or different sensors. One of the images is referred to the base image and the second image is referred as the sensed image. There are many ways to do image registration, including frequency domain methods and spatial domain methods. Frequency domain methods register the image in the transform domain (Persons et al., 2002). Spatial methods register the image in the image domain, matching intensity patterns or features in

images (Goshtasby, 2005). In this project, the image registration is done in the spatial domain. Image registration is described in the following.

Assume x and y are the coordinates of the base image and x' and y' are the coordinates of the same ground location in the sensed image, the mapping function connecting to these two coordinates is:

$$\begin{aligned} x' &= f(x, y) \\ y' &= g(x, y) \end{aligned} \tag{3-20}$$

f and *g* are high order polynomial mapping functions. Since the images needing registration in this project only have translation and rotation, the mapping function can be simplified as:

$$\begin{aligned} x' &= a_0 + a_1 x + a_2 y \\ y' &= b_0 + b_1 x + b_2 y \end{aligned} \tag{3-21}$$

The mapping function coefficients are calculated using control points X and Y:

$$\begin{aligned} a &= (X^T X)^{-1} X^T Y_x \\ b &= (X^T X)^{-1} X^T Y_y \end{aligned} \tag{3-22}$$

where $X = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_1 & x_2 & x_3 & \cdots & x_n \\ y_1 & y_2 & y_3 & \cdots & y_n \end{bmatrix}^T$ $\quad Y_x = \begin{bmatrix} x_1' \\ x_2' \\ \vdots \\ x_n' \end{bmatrix}$ $\quad Y_y = \begin{bmatrix} y_1' \\ y_2' \\ \vdots \\ y_n' \end{bmatrix}$

Here T is the transpose and n is the number of control points. After obtaining the coefficients of the mapping function, the image is finally registered with the process of geometric transformation. To obtain the control points, the scale-invariant feature transform (SIFT) (Lowe, 2004) is used to match the features between the base image and the sensed image. The coefficients of the mapping function can be calculated using the matched pairs. But the calculated coefficients are not accurate because not all the matched pairs are right. Therefore, an iterative algorithm, Random Sample Consensus (RANSAC) (Fischler et al., 1981), is used to

robustly estimate the coefficients of the affine transformation for image registration. These algorithms are described further below.

3.5.1 Scale-Invariant Feature Transform (SIFT)

SIFT, proposed by David G. Lowe in 1999, is an algorithm to extract and identify a large number of features (Lowe, 1999). Further, the method allows the descriptors to be reliably matched using a large database of features. To generate the set of image features, SIFT needs four major steps: scale-space extrema detection, keypoint localization, orientation assignment and keypoint descriptor respectively.

The first step is to detect interest points for SIFT features, which correspond to local extrema in the scale-space (Lowe, 1999). The scale space of an image can be defined as a function, $L(x,y,\sigma)$, that is the convolution of a variable-scale Gaussian with the input image. This is the process to simulate the multiple scale property of the image.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{3-23}$$

Here $I(x,y)$ is the input image and $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$ is the Gaussian function using $\sigma = \sqrt{2}$.

Lowe proposed the concept that using scale-space extrema in the difference-of Gaussian (DoG) function convolved with the image to detect stable key-point locations in the scale space. They can be computed from the difference of two nearby scales. The equation is shown below:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{3-24}$$

The initial image is incrementally convolved with Gaussians to produce images that differ by a constant factor k in scale space, where k is equal to $2^{1/s}$ and s is an integer number. Adjacent Gaussian images (left side of the Fig. 3-8) are subtracted from each other to produce the DoG

images (right side of the Fig. 3-8). After each octave, the Gaussian image is downsampled by a factor of 2 and the process is repeated.

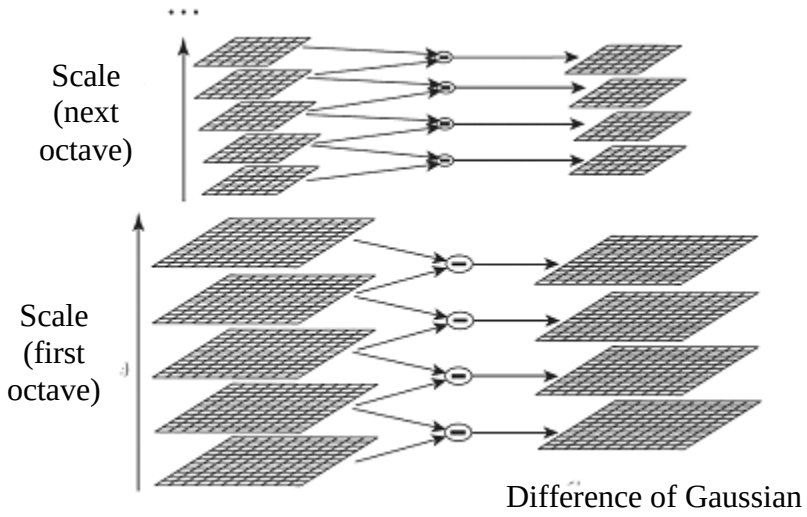


Figure 3-8: Illustration of the formation process of scale space and DoG
(From Lowe, 2004)

In the DoG image, each pixel, such as the one marked X in Fig. 3-9, is compared to its eight neighbors in the current image and nine neighbors in the scale above and below to decide the local maxima and minima of $D(x,y,\sigma)$, as shown in Fig. 3-9. If the pixel is a local maximum or minimum, it is selected as a candidate keypoint.

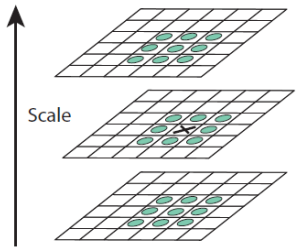


Figure 3-9: Illustration of the process of finding a candidate keypoint
(From Lowe, 2004)

The next step is to perform accurate keypoint localization. Points with low contrast and those that are poorly localized along an edge are rejected. Brown has developed a method to

perform keypoint localization (Brown et al., 2002). The method uses the Taylor expansion of scale-space function and shifts so that the origin is at the sample point.

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \tag{3-25}$$

Where $x=(x,y,\sigma)^T$ is the offset from this sample point. By taking the derivation of this function with respect to x and setting it to zero, the location of the extremum can be expressed as:

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \tag{3-26}$$

By substituting Equation 3-26 to Equation 3-25, the function value at the extremum, $D(\hat{x})$, can be obtained:

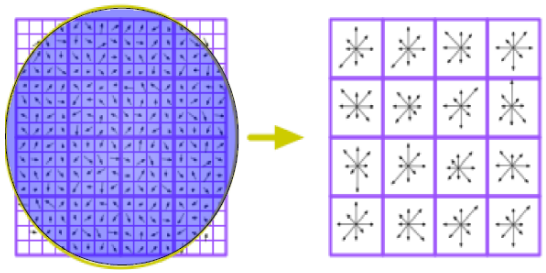
$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \tag{3-27}$$

All extrema with a value of $D(\hat{x})$ less than 0.03 are discarded.

To assign the orientation, a gradient orientation histogram is calculated in the neighborhood of the keypoint. The peak in the histogram corresponds to the dominant direction. This process is to test the rotation-invariant property of the keypoint.

The final step is to compute the keypoint descriptor. As is shown in Fig. 3-10 (a), a 16x16 window is regarded as the neighborhood of the keypoint. Each pixel has the gradient orientation and magnitude, which is indicated as the direction and length of the arrow respectively. In each 4x4 small window, eight orientation histograms are computed and accumulated. A descriptor is formed. The keypoint has 4x4 descriptor array and each contains 8 bin histograms. So a SIFT keypoint has 4x4x8=128 elements.

Euclidean distance was used to match the keypoints in the two images. First, the distances between one keypoint in the first image and each keypoint in the second image were calculated. Then in the second image, two points with the first nearest and second nearest distance were selected and the quotient (first nearest distance divided by the second nearest distance) was calculated. If the result was less than the specified threshold that is 0.6 in the Lowe's paper, the keypoint in the second image with the nearest distance was the matched point. Finally, repeat this process for all keypoints to find out all match pairs.



(a) Image gradient (b) Keypoint descriptor

Figure 3-10: Illustration of the computation of keypoint descriptor (From Jonas Hurreimann)

3.5.2 Random Sample Consensus (RANSAC)

RANSAC, a randomized estimator, was introduced by Fischler and Bolles (Fischler at al., 1981). This algorithm is commonly used to estimate the parameters of a model, using data that may be contaminated by outliers. RANSAC estimates a relation that fits the data, while simultaneously classifying data into inliers (points that fit the relation) and outliers (points that do not fit the relation). Due to its ability to tolerate a large fraction of outliers, this algorithm has been widely applied to estimation problems in computer vision, such as feature matching and registration.

RANSAC operates in a hypothesize-and-verify process. First randomly select the samples from the input data set. The model parameters are computed using the sample data. The size of

the sample depends on the model. For example, to calculate the affine transform parameters in image registration, three points are needed. In the next step, the quality of the model is evaluated on the full data set by calculating the error for each point. Then determine how many points from the set of all points fit within a predefined tolerance error ϵ. If the number of the inliers exceeds a predefined threshold τ, re-estimate the model parameters using all the identified inliers and terminate. If not, this hypothesize-and-verify loop is repeated until the criterion is met.

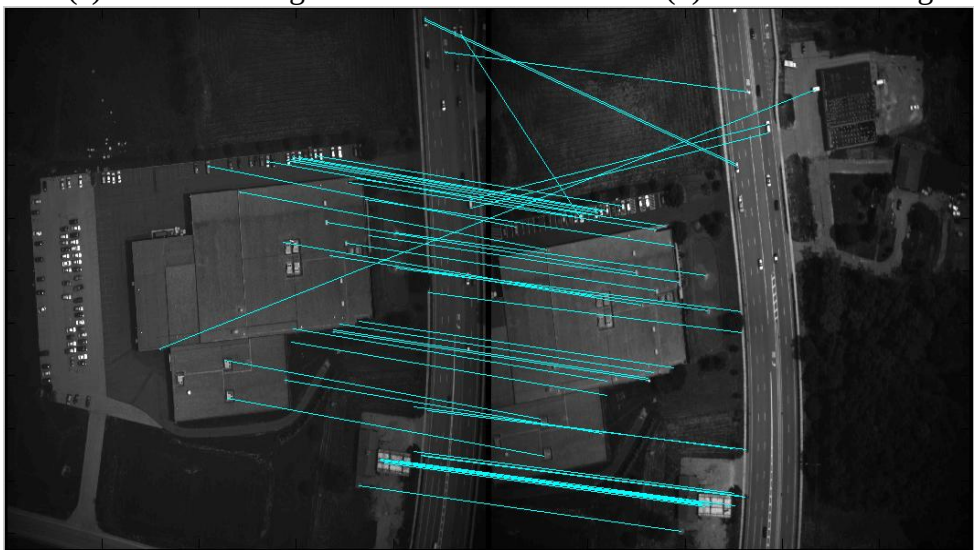
Two thresholds are used in the RANSAC operation. One is the error for each point to determine whether or not it agrees with the model. This threshold is equal to 5 in this thesis. Another is the number of inliers. This threshold is half of matched pairs that are different frame by frame. Take the frame 16 as an example, there are 55 matched pairs as listed in Fig. 3-11(a), so the threshold for this frame is 27. The advantage of RANSAC is its ability to do robust estimation of the model parameters, but it can only estimate one model for a particular data set. RANSAC may fail when more than one model exist in the data set.

3.5.3 Registration Result

For the MAPPS data, the first frame image is regarded as the base image; all other images are registered based on the first image. Take the frame 16 as the example; it is registered based on the first frame. In Fig. 3-11, (a) is the first base image, (b) is the sensed image (the sixteenth frame), (c) shows the matched pairs between (a) and (b) using SIFT, (d) is the common part of image (a) with image (b), and (e) is the registered image.

(a) The base image

(b) The sensed image



(c) SIFT matched image

(d) The common part of the base image with the sensed image

(e) The registered image

Figure 3-11: Illustration of image registration for MAPPS data

Not all the matched pairs are correct in Fig. 3-11 (c). When matching the keypoints in two images, the specified threshold (equal to 0.6 in Lowe's paper) is a criterion to decide whether two keypoints are the matched pair or not. The detailed procedure is discussed at the end of section 3.5.1. If the threshold is decreased, the number of matched pairs is less. Otherwise, the number of matched pairs is more. The number of wrongly matched pairs is also increased. However, these wrongly matched pairs do not affect registration result, because RANSAC is used to robustly classify matched pairs into inliers and outliers points.

To evaluate the result of image registration, the Peak Signal to Noise Ratio (PSNR) is used to quantitatively analyze. The PSNR term describes the similarity between two images and its unit is decibel (dB). The equation is defined as following:

$$PSNR = 10\log\frac{(2^n - 1)^2}{MSE} \tag{3-28}$$

Where n is the number of bits. The mean–squared error (MSE) represents the average squared difference between the digital counts from two compared images. The equation is defined as below:

$$MSE = \frac{\sum_{i=1}^{N_c}\sum_{j=1}^{N_r}(f_{ij} - \hat{f}_{ij})^2}{N_c N_r} \tag{3-29}$$

Where f_{ij} and $\hat{f}_{ij}$ are the digital counts of the two compared images and N_c and N_r are the number of columns and rows in both images.

Generally speaking, the greater the value PSNR, the smaller the difference between images, the more similar two images, and the better the quality of the registered image; contrarily, the smaller value, the greater difference is, two images are more different and the quality is poorer. Generally, the registered image has the great similarity with the base image when PSNR is larger

than 30. PSNR and MSE between image (e) and (d) are 34.2 and 24.7 respectively, which indicates the good performance of registration method.

4 Methods

For tracking vehicles it is necessary to detect the vehicles first, because we do not know the ground location for each vehicle. In this chapter, the target detection methods and the tracking algorithms are described.

4.1 Detection

Three target detection methods are used in this thesis. Local adaptive threshold method (Shafait et al., 2008) is applied to the MAPPS data set. Two other methods, RX (Reed et al., 1990) and change detection (Radke et al., 2005) are combined to detect the vehicles in the DIRSIG data set.

4.1.1 Local Adaptive Threshold

To segment the foreground objects, a threshold is commonly used. Pixels with intensity values above a threshold are considered as the foreground, and all the remaining pixels are the background.

$$D(x,y)=\begin{cases}1 & if\ \ g(x,y)\geq t(x,y)\\ 0 & otherwise\end{cases} \tag{4-1}$$

For the conventional threshold methods, the threshold *t(x,y)* is constant across the whole image, while the local adaptive threshold method varies the threshold dynamically across the whole image. For each pixel in the image, a threshold is calculated. The way to find the local threshold depends on the intensity values of the local neighborhood of each pixel. The mean or the median of the local neighborhood can be regarded as the threshold. The size of the neighborhood window has to be large enough to cover both foreground and background pixels; otherwise a poor threshold is chosen. In this thesis, the window size is about 60*60 pixels, although different frames have different window sizes.

When the mean or median value lies between the intensity values of foreground and background, they are separated easily. If the range of intensity values within a local neighborhood is very small, the mean of the local area is not suitable as a threshold, because its mean is close to the value of the center pixel. Finally, the threshold employed is not the mean, but (mean-C), where C is a constant value and can be negative or positive. All pixels in a uniform neighborhood are set to the background.

4.1.2 RX Detector

The RX detector (RXD) was developed by Reed and Yu to detect targets whose signatures are distinct from their background. This method is commonly used to detect the small targets for multi-spectral and hyperspectral images (Bartlet et al., 2011). The main idea of RXD is to use the sample covariance matrix to take into account the sample spectral correlation; it performs as the Mahalanobis distance.

$$R(x) = (x-m)^T \sum^{-1} (x-m) \tag{4-2}$$

where x is the pixel spectral vector, m is the mean spectral vector and $\sum$ is the sample covariance matrix of the image. Suppose that L is the number of spectral bands, then x is a L*1 column vector and $\sum$ is a L*L matrix. The equation illustrates that the RXD algorithm computes the Mahalanobis distance from each pixel to the spectral distribution of the global image. A large R(x) value corresponds to points that may be anomalous. Since the images produced by the RXD are generally grayscale, a threshold is needed to segment targets from the image background.

4.1.3 Change Detection

Moving target detection and segmentation can be achieved using change detection. Differencing the current frame from the previous frame, the difference image is formed. In

equation 4-3, $I_t(x)$ and $I_{t-1}(x)$ represent the current frame and the adjacent previous frame respectively. To threshold the difference image $D(x)$, the change mask $B(x)$ is generated according to the equation 4-4.

$$D(x) = I_t(x) - I_{t-1}(x) \tag{4-3}$$

$$B(x) = \begin{cases} 1 & if\ |D(x)| > \tau \\ 0 & otherwise \end{cases} \tag{4-4}$$

The threshold τ is chosen empirically. In this thesis, τ is mostly equal to 0.001 with the units of [W/cm^2/sr/μm], because DIRSIG typically generates spectral radiance imagery.

4.1.4 Combined Method

Both RX detection and change detection have shortcomings. For RX detection, non-moving man-made objects, such as buildings, may be detected as an anomaly. Therefore, some false alarms are produced using RX detection. For change detection, not all the background is stable from one frame to next frame, so some pixels are mistakenly considered as the foreground object. To overcome the disadvantages of both methods, the two binary images obtained from RX and change detection can be multiplied at the pixel level with the objective of suppressing false detections not present in both methods. This combined method obtains a good result because only anomalies in both binary images are likely to be selected as detections. Fig. 4-1 shows the process of the combined detection method. Frame t and frame t-1 are the adjacent frames at time t and t-1 respectively.

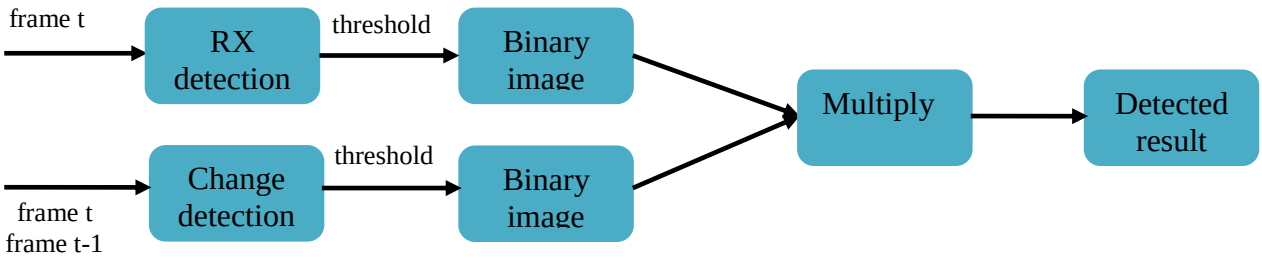


Figure 4-1: The flow chart of the combined detection method

4.2 Motion Tracking

Motion tracking depends on measurements of the locations and velocity to track vehicles. Fig. 4-2 is the flow chart of motion tracking used in this thesis. Assume there are N vehicles in the frame t, each vehicle is initialized with Kalman filter, and then predict the location in the next frame. Depending on the predicted location and the measurements, GNN (Global Nearest Neighbor) data association is applied to assign the measurements to the existing tracks. The measurements that are not assigned are regarded as new vehicles and initialized as new tracks. Finally, all the trackers predict the location in the next frame. In the following section, the Kalman filter and GNN algorithm are introduced.

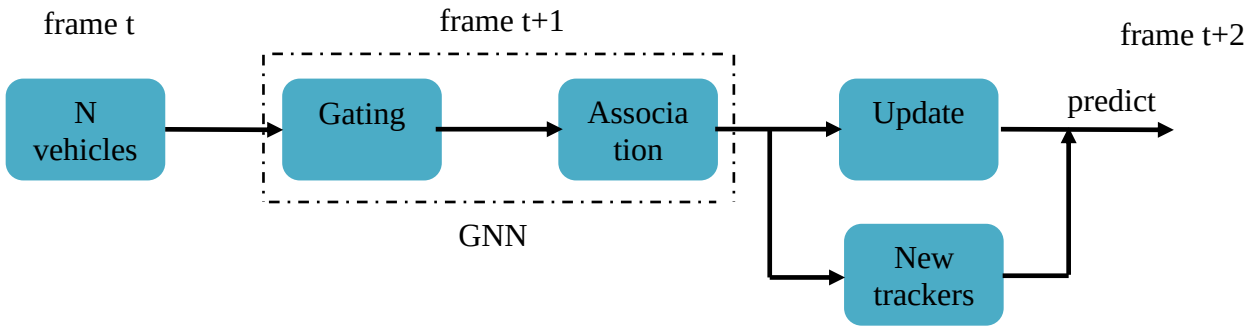


Figure 4-2: The flow chart of motion tracking

4.2.1 Kalman Filter

A Kalman filter (Maybeck, 1979) is used to estimate the state of a linear system. The state can refer to any measurable quantity, such as an object's location or velocity. The Kalman filter is a recursive two-stage filter. At each iteration, it is composed of a predict phase and an update phase.

The predict phase predicts the current location of the moving object based on the previous observation. We consider a tracking system where x_k is the state vector which represents the

dynamic behavior of the object, where subscript k indicates the discrete time. The equations for the predict phase are as following:

the system estimate:

$$x_k = Ax_{k-1} \tag{4-5}$$

the state prediction covariance:

$$P_k = A\,P_{k-1}A^T + Q \tag{4-6}$$

where, A is the state transition matrix, x_k and x_{k-1} represents the state at time k and k-1 respectively, Q is the noise covariance of the system. P is the predicted estimate covariance.

The update phase involves other system information, an observation as well as estimate calculations. When an observation z_k is made, the residual of that measurement is calculated as:

$$y_k = z_k - H\ x_k \tag{4-7}$$

H is the measurement matrix.

The optimal Kalman gain K_k is computed:

$$K_k = P_k H(HP_k H^T + R)^{-1} \tag{4-8}$$

Where, R is the observation noise covariance. Based on the above equation, the state update and the covariance update are computed as below:

$$x_{k+1} = x_k + K_k y_k \tag{4-9}$$

$$P_{k+1} = (I - K_k H)P_k \tag{4-10}$$

After the x_{k+1} and P_{k+1} is computed, they are used recursively to predict a new estimate. This recursive behavior of estimating the states is one of the highlights of the Kalman filter.

The system models of Kalman filters depend on a case by case analysis. The dimension of the system model can vary and may be 2, 4, 6 or other dimension including the position, velocity,

spatial size and so on. The system models of the Kalman filters for the two data sets considered here are different. The vehicles in the MAPPS data set move in two directions, nearly parallel. So the state vector x_0 is initialized using the top-left position of the tracking window (rectangular) as shown in Fig. 4-3(a), and velocity. The size of the tracking window is the same for all vehicles with the width 30 and height 50 pixels. For velocity, one method to initialize it is to use the difference between two positions from the first two consecutive frames. But it is convenient and effective to initialize with zero velocity and a large state covariance. The initialization of each parameter for MAPPS data is listed below respectively.

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad x_0 = \begin{bmatrix} x \\ y \\ x' \\ y' \end{bmatrix} \qquad P_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times 10^4$$

A is the state transition matrix, P_0 is the state covariance and x_0 is the state vector. Δt is the time interval between two adjacent frames and it is equal to 1 in MAPPS data. The measurement transition matrix H, the measurement noise covariance R and process noise covariance Q are listed below:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times 1/2$$

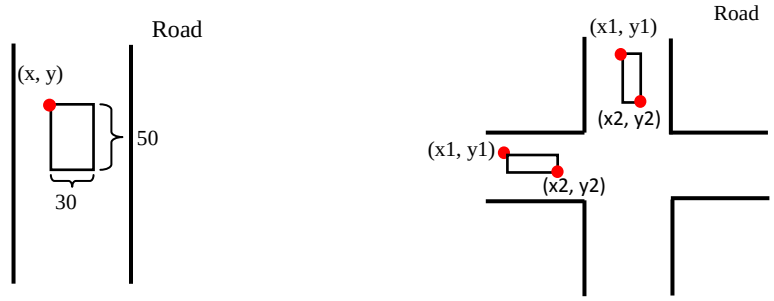
For the DIRSIG data set, the vehicles move in all directions. It is improper to set the size of tracking window of all the vehicles same. So the state vector used in DIRSIG data set is different from the one in the MAPPS data set. The state vector is six dimensional vector with the top-left and bottom-right locations of the tracking window and the velocity x' and y', as shown in Fig. 4-

3(b). Other parameters are also correspondingly changed. The total of six parameters are initialized as below:

$$A=\begin{bmatrix}1&0&0&0&\Delta t&0\\0&1&0&0&0&\Delta t\\0&0&1&0&\Delta t&0\\0&0&0&1&0&\Delta t\\0&0&0&0&1&0\\0&0&0&0&0&1\end{bmatrix} \quad x_0=\begin{bmatrix}x1\\y1\\x2\\y2\\x'\\y'\end{bmatrix} \quad P_0=\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\end{bmatrix}\times 10^3$$

$$H=\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\end{bmatrix} \quad R=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix} \quad Q=\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\end{bmatrix}\times 1/2$$

here, Δt is equal to 5 and x_0 is initialized with the detected location and zero velocity.



(a) MAPPS data (b) DIRSIG data

Figure 4-3: Illustration of initiation of Kalman Filter (red point indicates the initial position)

4.2.2 Global Nearest Neighbor Approach

In a multiple target tracking (MTT) system, data association is very important because the received measurements may not all arise from the real targets. Some of them may be from false alarms. Therefore, there always exist ambiguities in the association between the previous known

targets and measurements. Assigning a wrong measurement to an existing track often results in a lost track or wrong track.

Many data association methods are used in MTT systems, ranging from simple methods to complex ones, such as multiple hypotheses tracking (MHT) (Reid, 1979). The simple method, such as suboptimal nearest neighbor (Farina et al., 1985), can be easily implemented in MTT system, but its performance is degraded in a cluttered environment. The MHT method provides improved performance, but it is difficult to implement and a large number of hypothesis may be maintained. Global Nearest Neighbor method (Konstantinova et al., 2003) gives an optimal solution. Before associating the observation to tracker, gating is implemented first.

4.2.2.1 Gating

Gating (Blackman, 1986) is a coarse test for eliminating unlikely observation-to-track pairings. A gate is formed around the predicted position of a Kalman filter. All measurements within the gate are considered for updating this track. Actually, the measurement to update the track depends on the data association. The measurements outside the gating may be false alarms or newly appeared targets.

Using the Kalman filter, the state vector and the state covariance are predicted as in equations 4-5 and 4-6 respectively. The predicted measurement is:

$$z'_k = Hx_k \tag{4-11}$$

The residual vector between measured and predicted quantities is:

$$z'_{ij}(k) = z_j(k) - z_i'(k) \tag{4-12}$$

z_j is the j-th measurement, z_i is the i-th track predicted vector. The residual covariance matrix is computed:

$$S(k) = HP_kH^T + R \tag{4-13}$$

A validation region is defined as :

where G is the gate threshold. The $$d_{ij}^2 = z_{ij}^{'T} S_i^{-1} z_{ij}^{'} \le G \quad (4\text{-}14)$$

measurements within the gate or on the boundary of the gate are the validated measurements.

4.2.2.2 Association

Data association takes the output of the gating algorithm and makes the final measurement-to-track association (Bogler, 1989). When a single measurement lies in the gate of a single track, an assignment can be made immediately. But the conflict situation will arise when multiple measurements falls within a single gate or when a single measurement falls within the gates of several tracks. Fig. 4-4 illustrates the conflict situation. In the three predicted locations (P1, P2 and P3) for three trackers, corresponding gates are formed. Three measurements (O1, O2 and O3) fall in these gates. But the measurement O2 falls in the intersection of the gates of the three trackers. By calculating the distance from each measurement to each predicted location, the GNN method can decide which measurement should update which tracker.

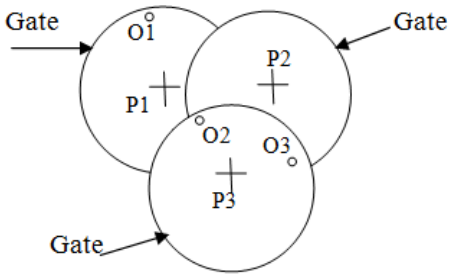


Figure 4-4: Illustration of the conflict situation

We assume there exist a set of n tracks and m measurements at the time index k, with m not necessary equal to n. A validation gate is defined by equation 4-14. The choice of G has to ensure that the correct measurements will lie within the gate. We define the cost matrix C (Konstantinova et al., 2003) for the assignment problem. The entries in matrix C are determined depending on equation 4-15. If measurement *j* is not in the gate of track *i*, the entry is set to 100;

otherwise, the entry is set to the squared Mahalanobis distance from the measurement *j* to the track *i*.

$$C_{ij} = \overbrace{\begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & \cdots & c_{1m} \\ c_{21} & c_{22} & c_{23} & c_{24} & \cdots & c_{2m} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ c_{n1} & c_{n2} & c_{n3} & c_{n4} & \cdots & c_{nm} \end{bmatrix}}^{j} \begin{matrix} 1 \\ 2 \\ \vdots \\ n \end{matrix} \Big\} i$$

$$C_{ij} = \begin{cases} 100 & \text{if measurement } j \text{ is not in the gate of track } i \\ d_{ij}^2 & \text{if measurement } j \text{ is in the gate of track } i \end{cases} \qquad (4\text{-}15)$$

The desired solution of the cost matrix is the one that minimizes the summed total distance. We use the Munkres algorithm (Munkres, 1957 and Bourgeois et al., 1971) to solve the assignment problem. But due to missed detections, it is possible that some tracks are associated with a measurement that is not in the gate of the track. So it is necessary to check the values of matrix C.

4.2.2.3 Munkres Algorithm

The Munkres method is an optimal method to solve the assignment problem. The goal of this method is to assign jobs to workers so as to minimize the total cost. This method has 6 steps:

Step 1: Create an N*M cost matrix in which each element represents the cost of assigning one of N workers to one of M jobs. The matrix can be rectangular or square. K=min(N,M).

Step 2: For each row, find the row minimum and subtract it from all entries on that row.

Step 3: For each column, find the column minimum and subtract it from all entries on that column.

Step 4: Draw lines across rows and columns to cover all zeros using the minimum number of lines.

Step 5: If the number of lines drawn is equal to K, the assignment is finished. If not, go to step 6.

Step 6: Find the smallest entry which is not covered by the lines, subtract it from each entry not covered by the lines, and add it to each entry which is covered by line. Then go back step 4.

To easily understand this algorithm, an example is used. Suppose there are three vehicles and three measurements, so K is equal to 3. In the cost matrix C_{ij}, 1, 2 and 3 are the minimum of each row respectively. Through step 2, the second matrix is obtained. 0, 1 and 2 are the minimum of each column in the second matrix. Subtract the column minimum, and the third matrix is formed. The minimum number of lines to cover all the zeros is 2, less than 3, so step 6 is implemented. In matrix four, 1 is the minimum entry that is not covered by the line. Subtracting it from each entry not covered by the lines, the fifth matrix is formed. Now draw lines again, three lines are needed to cover all zeros, so the assignment is finished. The measurements 3, 2 and 1 are used to update tracker 1, tracker 2 and tracker 3 respectively.

measurements j

$$C_{ij} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 & 2 \\ 0 & 2 & 4 \\ 0 & 3 & 6 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 2 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} \cancel{0} & \cancel{0} & \cancel{0} \\ 0 & 1 & 2 \\ 0 & 2 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 3 \end{bmatrix} \rightarrow \begin{bmatrix} \cancel{0} & \cancel{0} & \cancel{0} \\ 0 & 0 & 1 \\ 0 & 1 & 3 \end{bmatrix}$$

vehicles i

4.3 Feature Matching

In a complex environment, motion tracking often fails when vehicles split or vehicles are re-detected after missing detection. Motion tracking regards the detections that are not assigned using Munkres Algorithm as the new ones. This case is not always true because they may be the ones that existed before. Feature information, such as spectral measurements, can identify and

distinguish the vehicles because different materials have different spectral information. So spectral histogram based tracking (Nguyen et al., 2010 and Brown et al., 2006) is widely used for object tracking.

The process of spectral feature matching is shown in Fig. 4-5. Assume there are n targets in the frame t and a rectangular target region identifies each target. The spectral feature in each target region is extracted and a histogram is computed. In the next frame, m new targets appear and the corresponding features are also extracted. The features of each new target are compared with the features of prior frames to check whether the new appeared target is a new one or the one existed before. If the minimum of compared results is above the threshold, this target is regarded as the new target, otherwise, this target is marked as an existing target which has been re-detected again.

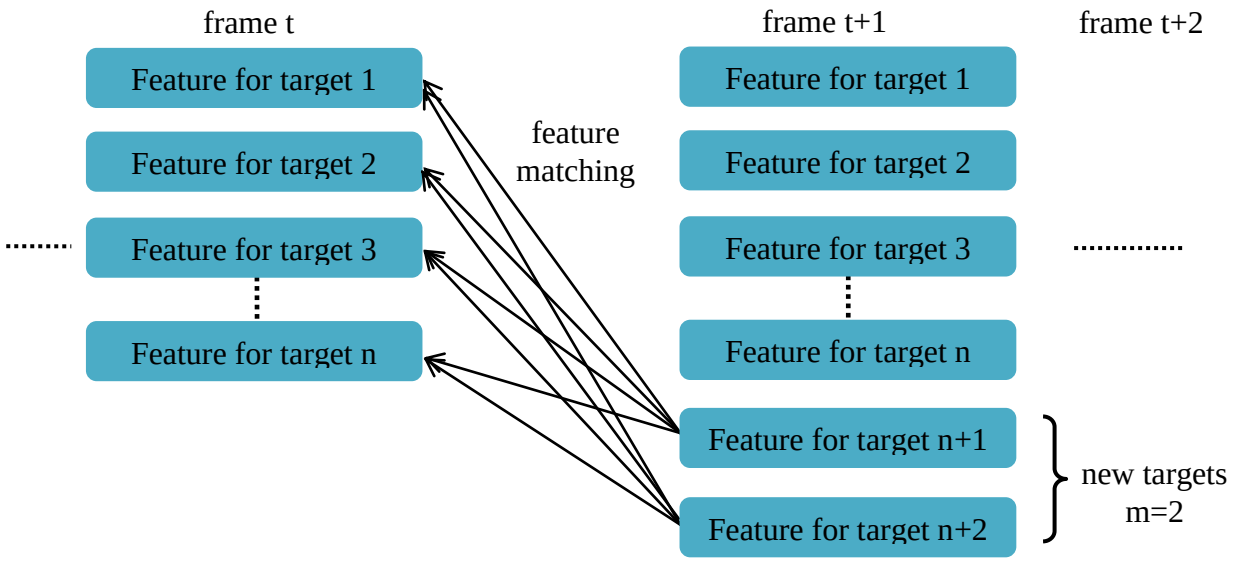


Figure 4-5: The illustration of feature matching

The statistical measure, Bhattacharyya measure (Djouadi et al., 1990 and Kailath, 1967), is used to calculate the similarity between histograms. The Bhattacharyya coefficient $\rho(p, p')$ measures the amount of relative closeness between two statistical samples:

$$\rho(y) = \rho(p(y), q) = \sum_{u=1}^{Nm} \sqrt{p_u q_u} \tag{4-16}$$

Where p and q represent the spectral probability density function (spdf) of the reference object and the candidate object respectively, each consisting of Nm bins with respective probability p_u and q_u. N is the number of bands and m is the number of bins for each band.

The distance between two histograms is defined as:

$$d_{Bhat}(y) = \sqrt{1 - \rho(y)} \tag{4-17}$$

When d_{Bhat} is equal to 1, it means the maximum mismatch happens; when d_{Bhat} is equal to zero, it means two histograms have maximum match.

4.4 Feature Aided Tracking

Motion tracking regards all the detections that are not used to update trackers as new ones. Actually, they are not always new ones. So motion-only tracking will assign the wrong track for some targets. Adding features into tracking will solve this problem. Since the illumination may slightly vary between frames, the histogram may be not same for each target in all frames. In this thesis, the features of each target at each frame are stored in order to improve the matching accuracy.

Each target has four properties, position, velocity, identification and feature. For the second frame (this is the first tracking frame owing to use change detection), the position is initialized using the detected result and the velocity is initialized to zero for each target. Also identification is marked and features extracted for each target. Assume in frame t, there are n detected targets and each one has been assigned a tracker. In the frame t+1, m targets are detected and the corresponding features extracted. Depending on the GNN data association, some detections are

used to update the n trackers. The features of the remaining detections are compared with all the features of prior frames to check whether they are the targets that existed before or new ones. The Bhattacharyya distance is used to measure the similarity between features. If the minimum Bhattacharyya distance among a group of distances is above the threshold, this target is a new one; otherwise, it is corrected with the prior tracking ID. The threshold used in this thesis is 0.447, which means the similarity of features should be at least 80%. If there is no detection to update one track in three-adjacent frames, this track is given up. Fig. 4-6 is the flow chart of the feature-aided tracking algorithm.

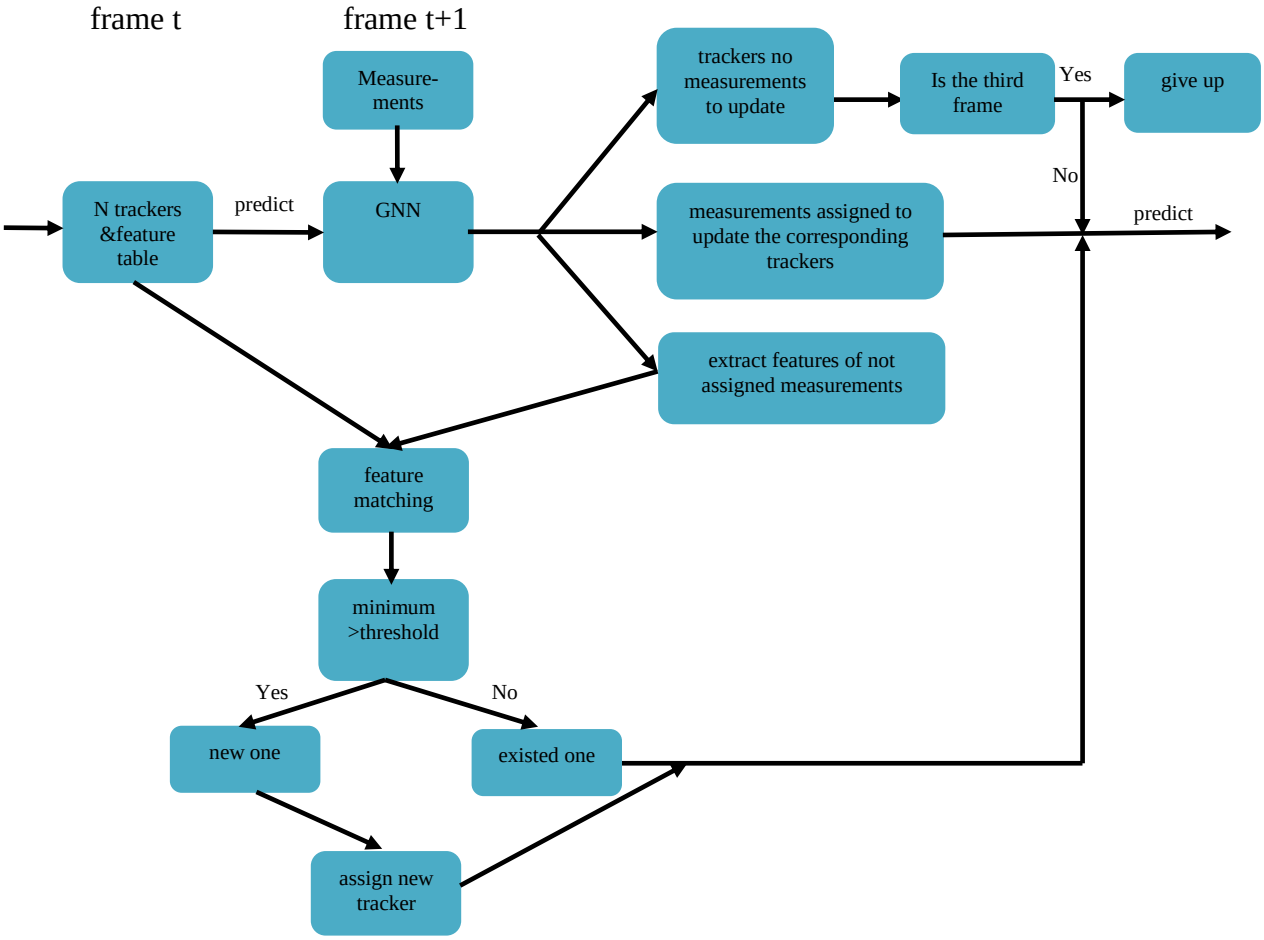


Figure 4-6: The flow chart of feature-aided tracking

4.5 Metrics for Performance Evaluation

Performance evaluation metrics aim to quantitatively analyze the tracking performance based on system results and ground truth data. In most cases, system results could differ from ground truth because of possible errors. Comparing system results and ground truth, four outcomes are possible. These are true positives, false positives, true negatives and false negatives. In a tracking problem, true positives are typically named "correct tracks", false positives are "false alarms", true negatives are ones that switch to other tracks and false negatives are "lost tracks". Based on these four classifications, the metric, Multiple Object Tracking Accuracy (MOTA), can be computed.

$$MOTA = 1 - (\overline{m} + \overline{fp} + \overline{me}) \tag{4-18}$$

where m_t, fp_t and me_t are the number of misses, of false alarms and of ones that switch to other tracks respectively at time t. g_t is the ground truth for time t. The MOTA can be seen as subtracting 3 error ratios from 1.

$$\overline{m} = \frac{m_t}{g_t} \tag{4-19}$$

$\overline{m}$ is the ratio of misses in the sequence, computed over the total number of objects present in all frames. The ratio of false alarms and the ratio of mismatches are calculated as in the below equation in a similar way.

$$\overline{fp} = \frac{fp_t}{g_t} \tag{4-20}$$

$$\overline{me} = \frac{me_t}{g_t} \tag{4-21}$$

The MOTA accounts for all target configuration errors made by the tracker. It can give a very intuitive measure of the tracker's performance, the larger ratio of false alarms, mismatches and misses, the smaller MOTA it is.

Another metric, called "track completeness", is the ratio of the number of valid tracks that are indeed moving vehicles to the total number of moving vehicles within the area of interest that should be tracked. This metric is computed at each time epoch and determines the performance of the target tracker.

$$Track\,Completeness = \frac{\#of\ Valid\ Tracks}{\#of\ "Should"Tracks} \tag{4-22}$$

The "track purity" metric evaluates whether a valid track stays with a single vehicle or skips to another vehicle for one or more time epochs. This metric is calculated target by target. An aggregate track purity metric averages all vehicle track purities to provide a single scalar number as a simulation performance indication (Presnar, M. D., 2010).

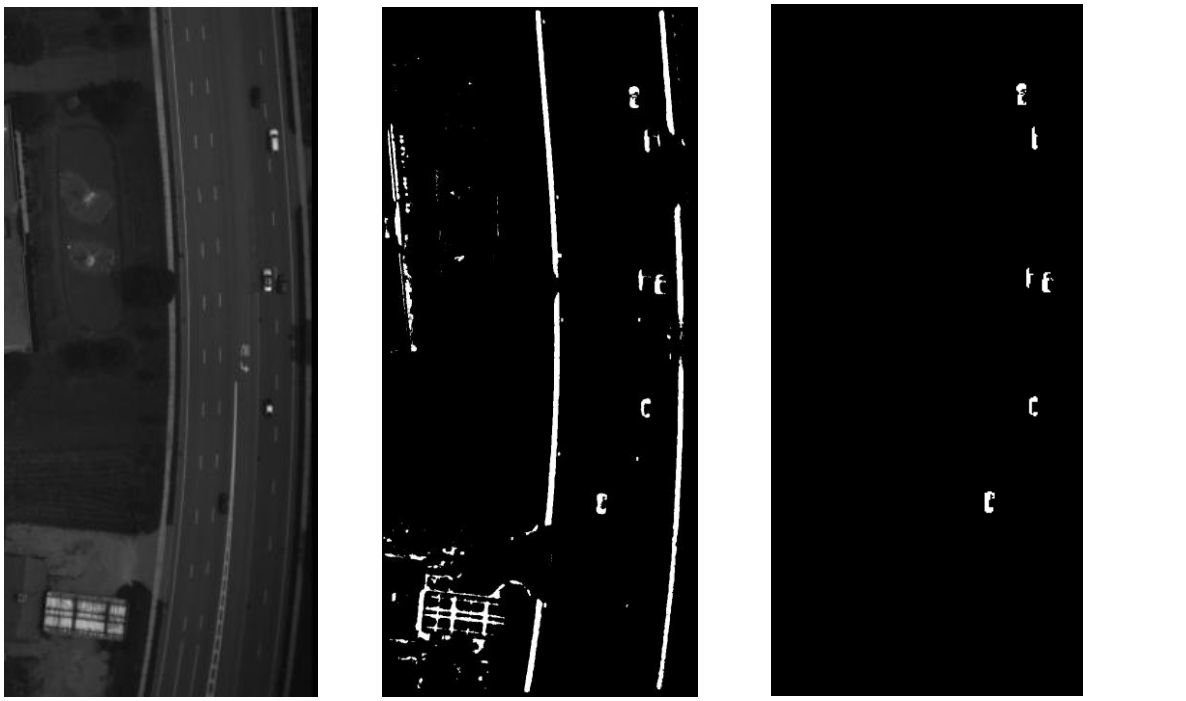
$$\text{Track Purity} = \frac{\#\text{of epochs a valid track maintained the same truth vehicle}}{\text{Total}\#\text{of epochs in a valid track}} \tag{4-23}$$

5 Results

In this chapter, the detection and tracking results are shown respectively. Further, the performance metrics of detection and tracking algorithms are calculated and analyzed.

5.1 Detection Results of MAPPS Data

The local adaptive threshold method is applied to MAPPS data set to detect the vehicles. Two parameters are set in the algorithm for each frame. One parameter is the window size regarded as the local neighborhood for calculating the mean. Another parameter is the constant value C in order to segment all pixels which exist in a uniform neighborhood (*e.g.* along the margins) are set to background. Although the values of the parameters are different frame by frame, the window size is about 60x60 and the C is about 18. The detected results of several frames are shown in the figures below.

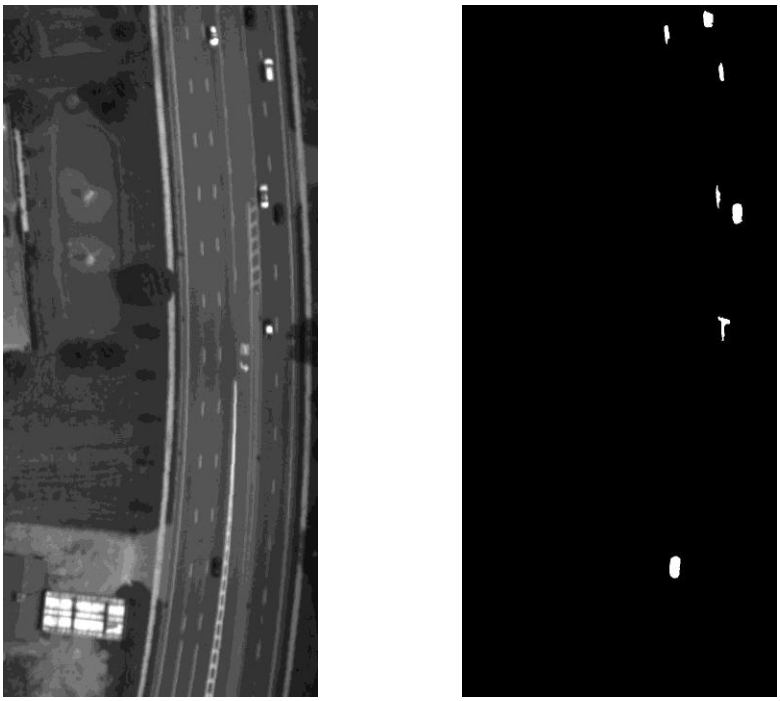


(a) Panchromatic image (b) Detected result (c) Detected result after spatial filter

Figure 5-1: Detected result using local adaptive threshold for MAPPS frame 1 data

(a) Panchromatic image (b) Detected result after spatial filter

Figure 5-2: Detected result using local adaptive threshold for MAPPS frame 5 data



(a) Panchromatic image (b) Detected result after spatial filter

Figure 5-3: Detected result using local adaptive threshold for MAPPS frame 6 data

After using local adaptive threshold method, spatial filter is applied to binary image to further eliminate false alarms again. The size of spatial filter depends on the estimation of the target size. In this paper, the width of the spatial filter ranges from 5~38, and the height ranges from 30~80 pixels. Fig. 5-1 (b) shows the detection results without using spatial filter and (c) is the result after using spatial filter.

As shown in the detected results of the above three figures, there are some missing pixels in the middle of the vehicles. But this case does not affect the tracking result, only the maximum and minimum of X coordinates and Y coordinates of the boundary are saved as the location for each vehicle. In Fig. 5-2(a), the vehicle marked with a red rectangle has been missing detected from frame 5. In frame 6, this vehicle is re-detected again, as shown in Fig. 5-3(b). The detection results of other frames are described in the below table.

Table 1: Illustration of the detection result for each frame of MAPPS data

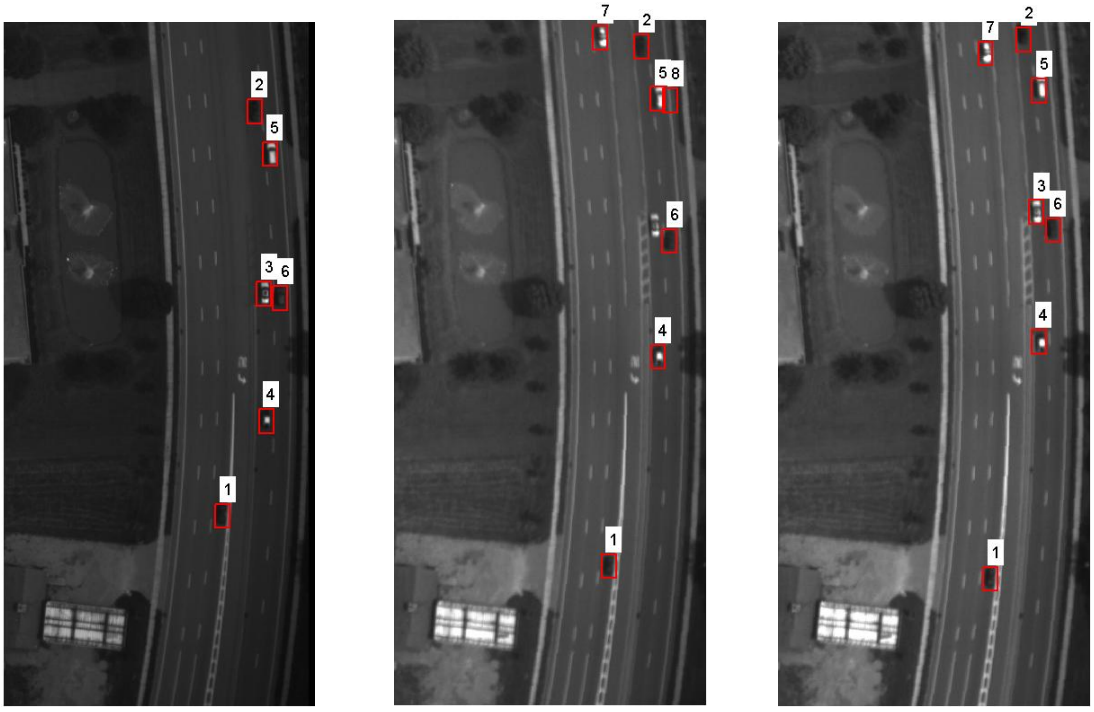
Frame Number	Vehicle Number	Detected Number	Miss Detect	False Detect		Frame Number	Vehicle Number	Detected Number	Miss Detect	False Detect
1	6	6	0	0		9	6	6	0	0
2	6	6	0	0		10	6	6	0	0
3	6	6	0	0		11	6	6	0	0
4	6	6	0	0		12	6	6	0	0
5	7	7	1	1		13	6	5	1	0
6	7	7	0	0		14	6	5	1	0
7	6	6	0	0		15	6	5	1	0
8	6	6	0	0		16	6	5	1	0

5.2 Tracking Results of MAPPS Data

5.2.1 Results of Feature-aid Tracking

For MAPPS data, some vehicles are tracked using the method of feature-aided tracking, but some vehicles are tracked using motion tracking only. This special case results from the property of data in that each frame of the data set is a one band one polarization panchromatic image. To use the multispectral information, each vehicle must be registered independently. Although the data set has six spectral bands, only the sequential four spectral bands are used due to the fast

speed of vehicles relative to the data acquisition speed. Further, most vehicles are out of the frame by the time these four spectral bands are collected, so only four vehicles are registered. The initiation of the first frame is shown in Fig. 5-4(a). The vehicles with the identity 1, 3, 4, 6 are tracked using the feature-aided method. All other vehicles are tracked using motion tracking. The tracking results of several frames have been shown in the figures below.



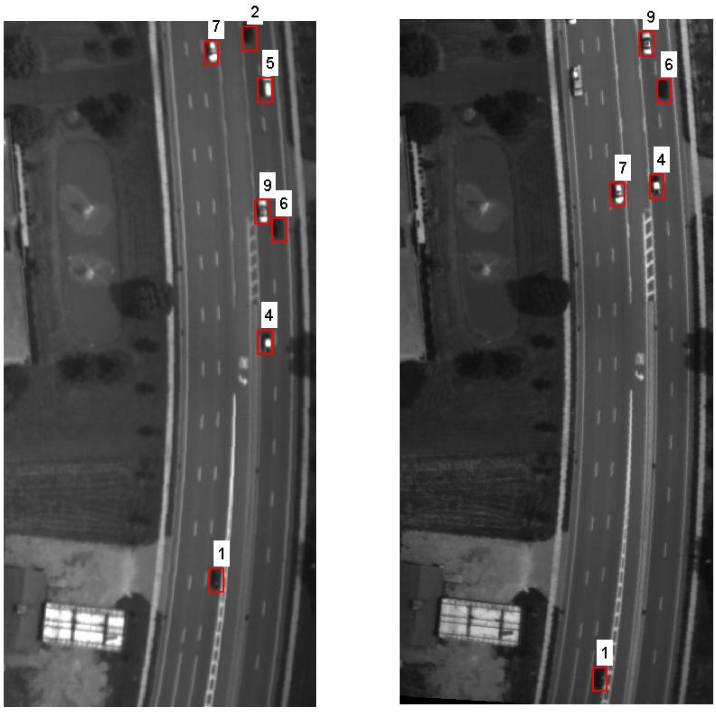
(a) Frame 1 (b) Frame 5 (c) Frame 6

Figure 5-4: Feature-aid Tracking Results for MAPPS data

Vehicles with the identity 1, 3, 4 and 6 are tracking using feature-aided tracking. A special case is vehicle 3. As was not detected in frame 5 shown in the Fig. 5-2. Motion-only tracking will lose this vehicle. In frame 6, this vehicle is re-detected again. Motion-only tracking will regard this vehicle as a new one, but the feature-aided method can recognize it using the feature matching. It is actually the one that existed before.

5.2.2 Results of Motion-only Tracking

Figure 5-5 illustrates the tracking results of motion-only tracking. Motion-only tracking cannot recognize the re-detected vehicle and will regard it as a new vehicle and assign a new tracker. Take the vehicle 3 as the example. It is not detected in frame 5 and re-detected in frame 6. Unlike the feature-aid tracking, motion-only tracking regards the redetected vehicle 3 as a new vehicle and assigns the new tracker (ID 9), as show in Fig. 5-5 (a). In the latter frames, this vehicle is tracked as vehicle 9 until the last frame.



(a) Frame 6 (b) Frame 16

Figure 5-5: Motion-only Tracking Results for MAPPS data

5.2.3 Performance Metrics for MAPPS Data

● Track Completeness

Track completeness is the ratio of the number of valid tracks to the number of vehicles that should be tracked. This metric reflects the condition of missing detection and losing track for

each frame. Perfect track completeness is equal to 1, which means no missed detections and track loss. Otherwise, missed detection or/and lost tracks happen.

Table 2: Track completeness for each frame of MAPPS data

Num. Frame	Track completeness	Num. Frame	Track completeness
1	1	9	1
2	1	10	1
3	1	11	1
4	1	12	1
5	0.86	13	0.83
6	1	14	0.83
7	1	15	0.83
8	1	16	0.83

● Track purity

Track purity, calculated target by target, is to evaluate whether a valid track stays with a single vehicle or skips to another one for one or more time epochs. Table 3 shows the track purity for both feature-aid tracking and motion-only tracking. Comparing track purity of feature-aid tracking with that of motion-only tracking for each vehicle, most vehicles have the equal track purity for both tracking methods. But if the vehicle is missed detected and re-detected again, feature-aid tracking can obtain better performance than motion-only tracking, as was the case for vehicle 3 shown in the table. In addition, the ID of the same vehicle, such as vehicle 8, may be

Table 3: Track purity for both feature-aid and motion tracking for MAPPS data

Num. vehicle	Feature-aid tracking		Motion-only tracking	
	ID of Vehicle	Track Purity	ID of Vehicle	Track Purity
1	1	1	1	1
2	2	1	2	1
3	3	15/16	3	4/16
4	4	1	4	1
5	5	1	5	1
6	6	1	6	1
7	7	1	7	1
8	9	2/6	10	2/6

different in feature-aid and motion tracking. The reason is that motion tracking regards a redetected vehicle as a new vehicle and assigns a new ID, but feature-aid tracking can recognize the vehicle and assign the ID using the prior one.

● Multiple Object Tracking Accuracy

Track completeness and track purity metrics evaluate the performance of valid tracks, while MOTA considers both valid tracks and invalid tracks (false alarms). Take frame 5 as the example, MOTA of frame 5 is equal to 0.71, which is less than track completeness value 0.86 in the same frame. The reason is MOTA considers both the false alarms and detection-miss.

Table 4: MOTA for both feature-aid and motion tracking for MAPPS data

Num. Frame	MOTA for motion tracking	MOTA for feature-aid tracking	Num. Frame	MOTA for motion tracking	MOTA for feature-aid tracking
1	1	1	9	1	1
2	1	1	10	1	1
3	1	1	11	1	1
4	1	1	12	1	1
5	0.71	0.71	13	0.83	0.83
6	0.86	1	14	0.83	0.83
7	1	1	15	0.83	0.83
8	1	1	16	0.83	0.83

Comparing MOTA of feature-aid tracking with that of motion tracking, mostly MOTA values are the same except for those of frame 6. The MOTA of feature-aid tracking of this frame is equal to 1, but 0.86 for motion tracking. The reason is that feature tracking can recognize the re-detected vehicle, but motion tracking cannot.

By averaging the MOTA values of feature-aid and motion tracking, the average value of feature-aid tracking is 0.94 and that of motion tracking is 0.93. In other words, feature-aid tracking increases MOTA 1 percent comparing with that of motion tracking.

Similarly, by averaging track purities, the average value of feature-aid tracking is 0.91, but that of motion tracking is only equal to 0.82. Therefore, we can conclude that feature-aid tracking can get better performance than motion only tracking.

5.3 Detection Results of DIRSIG Data

This session includes the detection results for DIRSIG data with and without trees respectively. Also noise is added into the data sets to analysis how the noise affects detection results. Combining RX with change detection method is applied to each case to detect vehicles.

5.3.1 Detection Results of Data with Trees

● Results of without noise case

RX detection algorithm is applied to spectropolarimetic DIRSIG data first. Fig. 5-6 (a) and (b) show the RGB image of the twenty-second frame and the corresponding detected result by RX detection using 24 spectropolarimetic images respectively. There are a lot of false detections in the result, because all of the man-made objects may be detected as anomalies. Then change detection is applied to detect the moving targets in the image by differing two adjacent frames. Fig. 5-6 (c) shows the result using change detection by differing the twenty-second frame with the twenty-first frame. A lot of false alarms are produced because not all the background is stable in two frames. Finally by combining RX detection with change detection, false detections are eliminated effectively. This good detection result is shown in Fig. 5-6 (d).

(a) RGB image

(b) Detected result of RX detection

(c) Detected result of change detection

(d) Final result using combined method

Figure 5-6: Detected results of the twenty-second frame of DIRSIG data with trees (without noise)

● Results of with Noise case

Noise affects detection results. Some vehicles are missing detected in the noise data set. Fig. 5-7 (b) shows the detected results of the twenty-second frame when SNR is equal to 30. Comparing with the detected result shown in Fig. 5-6 (d), several vehicles are missing detected in the noise data set.

(a) RGB image with noise (SNR=30)

(b) Final result using combined method (with noise SNR=30)

Figure 5-7: Detected results of the twenty-second frame of DIRSIG data with trees (with noise)

5.3.2 Detection Results of Data without Trees

● Results of without noise case

Vehicles are not detected when they are fully covered by trees. Therefore, the data set without trees is used in my project to evaluate the performance of detection and tracking method

used in the uncovered situation. The parameters of the data without trees are the same as those of the data with trees. Fig. 5-8 shows the detection results for the second frame with and without trees respectively. Several vehicles are fully covered by trees so they are missing detected, as shown in Fig. 5-8 (c). But they are all detected in the data set without trees.

(a) RGB image with trees (Frame 2)

(b) RGB image without trees (Frame 2)

(c) Detection result with trees

(d) Detection result without trees

Figure 5-8: Comparison of the detected results of the second frame of DIRSIG data (without/ with trees)

● Results of with noise case

Noise is added into the data without trees to investigate how the noise affects the detection results. Fig. 5-9 (a) is the RGB image with noise (SNR=10). Fig. 5-9 (b) is the corresponding detected result. Comparing Fig. 5-9 (b) with the (d) shown in Fig. 5-8, there is a vehicle missing

detected in the noise data set. Therefore, the detection performance decreases under the effect of noise.

(a) RGB image with noise (SNR=10)

(b) Detected result with noise (SNR=10)

Figure 5-9: Detected result of the second frame of DIRSIG data without trees (with noise)

Table 5 shows the detection results of DIRSIG data in different cases. Although the data with trees has 23 frames, only the results of 11 frames are shown in Table 5 for comparison because the data without trees has 11 frames. The average percent detection is calculated based on Table 5 for comparing the detection performance in different case.

$$\text{Average percent detection} = \frac{\sum_t (\#\text{of detection/ground truth})}{\#\text{of frames}} \tag{5-1}$$

In the case of without trees and noise, the detection method can get the best results, 95 percent detection. When noise is added into the data without trees, the detection performance is affected by noise, lower 5 percent than that of without noise case. In the DIRSIG data with trees, the detection performance is 71 percent due to occlusion. When noise is added into data, the performance is bad only 66 percent.

Table 5: The detection results of DIRSIG data

Num. Frame	Ground truth	Num. of detection without trees and noise	Num. of detection without trees and with noise	Num. of detection with trees and without noise	Num. of detection with trees and noise
2	12	12	11	9	7
3	13	13	11	10	7
4	13	13	12	8	8
5	17	12	12	10	12
6	17	15	14	9	8
7	17	16	15	12	10
8	17	17	15	13	14
9	17	17	17	14	13
10	17	17	17	13	12
11	17	17	17	14	14

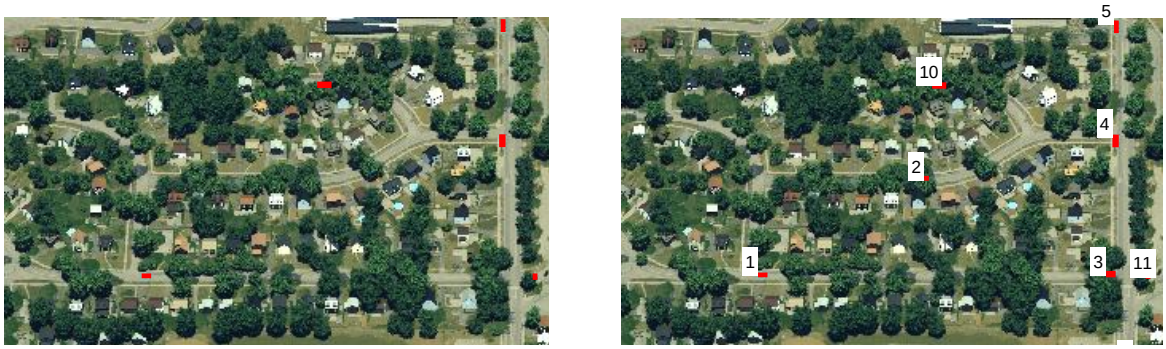
5.4 Tracking Results of DIRSIG Data

5.4.1 Three Frame Give-up Tracking

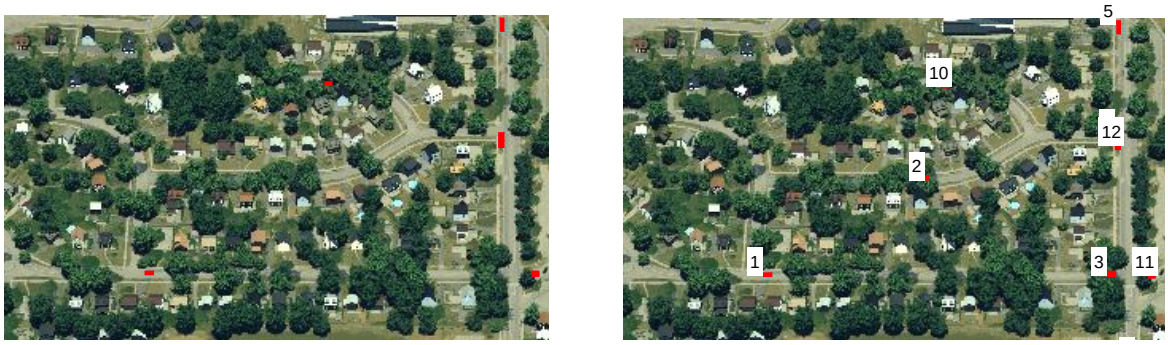
In frame t, each target possesses a Kalman filter to predict its location in frame t+1. Depending on the predicted locations and measurements, GNN data association can assign which measurement should update which Kalman filter. If no measurement is assigned to update a tracker, this tracker will give up tracking the corresponding target from frame t+1. Suppose in frame t+2, this target has been detected again and will be regarded as a new one. It is necessary to compare its feature with the prior features to check whether it is really a new one or the one that is miss detected. This one frame give-up tracking costs computation time.

Three frame give-up tracking can reduce the times of feature matching procedure. Assume in frame t+1, there is no measurement to update a tracker, this tracker doesn't be given up and predicts its location in frame t+2. Even there is no measurement to update it in frame t+2, it doesn't be given up and predicts its position in frame t+3. In this frame, the tracker may be deleted or continues tracking depending on whether there exists a measurement to update it.

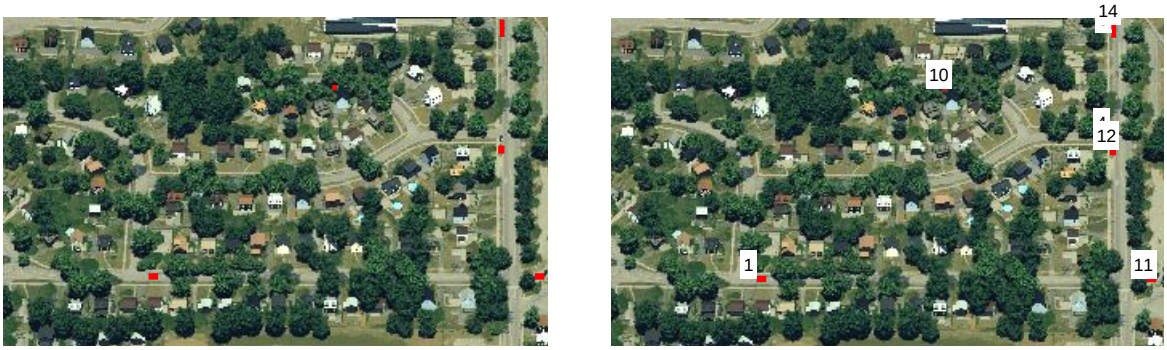
(a) Marked detection (left) and tracking result (right) of a part of frame 3



(b) Marked detection (left) and tracking result (right) of a part of frame 4



(c) Marked detection (left) and tracking result (right) of a part of frame 5



(d) Marked detection (left) and tracking result (right) of a part of frame 6

Figure 5-10: Illustration of the case of three frame give-up tracking

Fig. 5-10 shows the case of three-frame give-up tracking. Targets 2 and 3 are detected in frame 3 and missing detected in the following frames (4, 5 and 6). The trackers give up these two targets until there are no corresponding measurements to update them in the frame 6, as shown in the right of Fig. 5-10 (d).

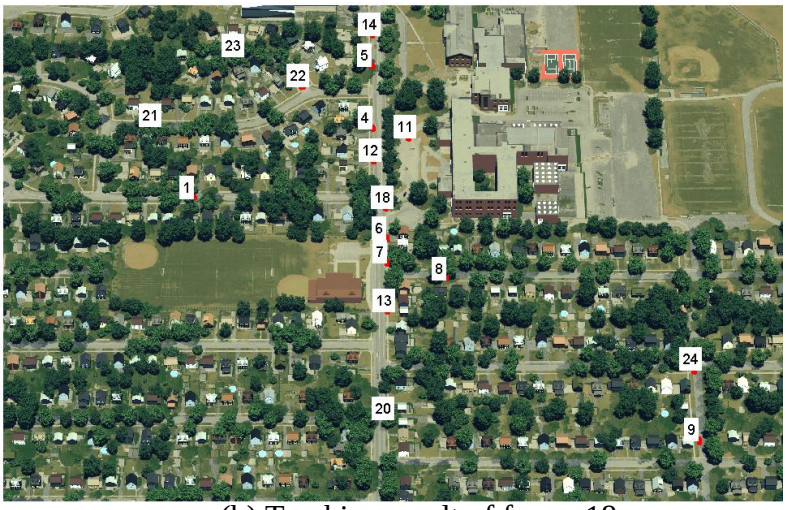
Comparing three frame given-up with one frame given-up tracking, the former has two advantages over the latter. On the one hand, three frame given-up tracking improves the tracking performance because it predicts the location of the target that has been missing detected for two consecutive frames. On the other hand, three frame given-up tracking reduces the times of feature matching, as described before.

5.4.2 Tracking Results of DIRSIG Data with Trees

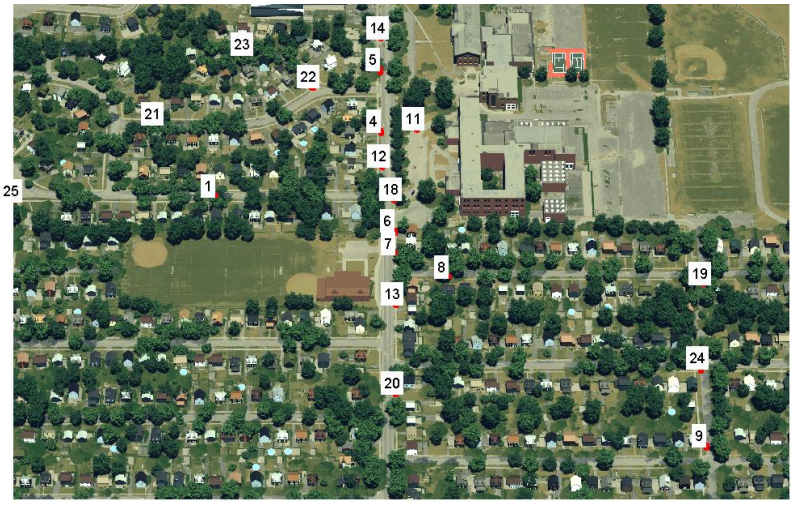
● Feature-aid Tracking Results

Motion-only tracking will regard the measurements that are not used to update the trackers as the new ones. But feature-aided tracking compares features with the prior features to correct tracking identification. The prior features are the collection of features of each target extracted in each frame. Fig. 5-11 illustrates the condition of correcting the tracking ID. Target 19 is detected and tracked from frame 10. It is miss detected from frame 16 and given up at frame 18. In frame 19, it is re-detected again. The feature-aided tracking can recognize that it is target 19 instead of a new target.

(a) Tracking result of frame 10



(b) Tracking result of frame 18



(c) Tracking result of frame 19

Figure 5-11: Illustration of correcting tracking ID for a re-detected target

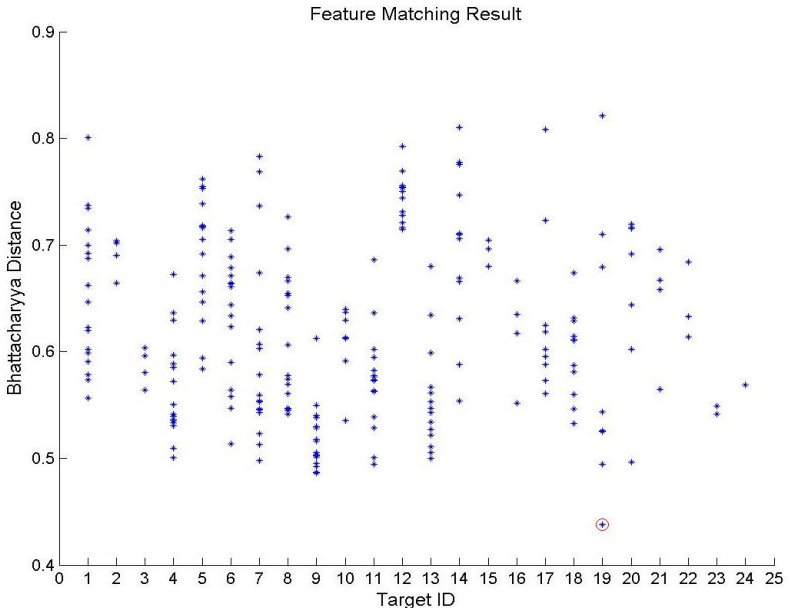


Figure 5-12: Result of spectral feature matching for a re-detected target

Fig. 5-12 shows the feature matching results of a re-detected target with the prior features. The X coordinate indicates there are 24 targets in the prior frames. The Y coordinate shows the Bhattacharyya distance between the features of re-detected target and the feature of each target at each prior frame respectively. The red circle marks the minimum distance which corresponds to the target 19, which indicates the re-detected target actually is the target 19.

● Motion Tracking Results

Motion tracking regards the re-detected vehicles as the new vehicles and assigns a new track ID. Also take the vehicle 19 as the example. Vehicle 19 is lost track in frame 18 and re-detected in frame 19. Motion tracking regards it as the new vehicle and assigns the new track 26, as shown in Fig. 5-13.

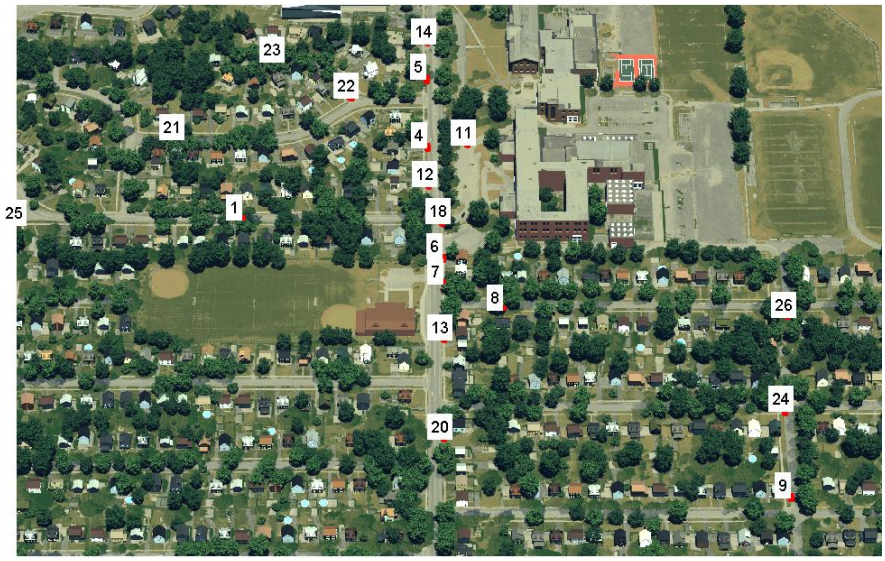


Figure 5-13: Tracking result of frame 19 using motion tracking

- Results of with noise case

Noise affects the detection and tracking results. For the DIRSIG data with trees, noise with SNR equal to 30 is added into data. Fig. 5-14 shows several frames with and without noise. The images on the left side are the tracking results of data with noise, the ones on the right side are the results of data without noise. Comparing image (a) with (b), two vehicles are miss detected in the data with noise. In frame 15, the result with noise has one missed detection and one false alarm, as does the frame 23. In addition, the order of vehicle ID in the data with noise is different from that of data without noise, because the detection sequences of vehicles are different in the two data sets.

(a) Tracking result of frame 2 with noise

(b) Tracking result of frame 2 without noise

(c) Tracking result of frame 15 with noise

(d) Tracking result of frame 15 without noise

(e) Tracking result of frame 23 with noise

(f) Tracking result of frame 23 without noise

Figure 5-14: Tracking results for DIRSIG data with trees (with/without noise)

5.4.3 Performance Metrics for DIRSIG Data with Trees

Table 6 lists the comparison of the track completeness for each frame of DIRSIG data with trees under with/without noise case. Track completeness, expressed as a fraction, is the ration of the number of valid tracks to the number of vehicles that should be tracked. The denominator indicates the ground truth and the numerator is the number of valid tracks. In without noise case, the track completeness of two frames, frame 17 and 18, is equal to 1. All others are less than 1 because of missed detection. Comparing the track completeness of with noise with that of without noise case for each frame, the former is worse than the latter because of the effect of noise.

Table 6: Track completeness for each frame of DIRSIG data with trees(with/without noise)

Num. Frame	Track completeness Without noise	Track completeness With noise
2	9/12	7/12
3	10/13	7/13
4	11/13	8/13
5	13/17	12/17
6	12/17	13/17
7	12/17	13/17
8	15/17	13/17
9	15/17	14/17
10	15/17	15/17
11	15/17	14/17
12	15/17	15/17
13	15/17	15/17
14	15/17	14/17
15	17/17	16/17
16	17/17	15/17
17	17/18	16/18
18	17/19	16/19
19	19/20	17/20
20	18/20	18/20
21	19/21	18/21
22	18/21	17/21
23	18/21	16/21

Table 7: Track purity for both feature-aid and motion tracking for DIRSIG data with trees

Without noise				With noise		
ID of vehicles	Track Purity for feature-aid tracking	ID of vehicles	Track Purity for motion tracking	ID of vehicles	Track Purity for feature-aid tracking	Track Purity for motion tracking
1	1	1	1	1	14/22	14/22
2	4/22	2	4/22	16	3/16	3/16
3	4/22	3	4/22	17	9/16	9/16
4	1	4	1	2	1	1
5	1	5	1	3	1	1
6	1	6	1	4	6/22	6/22
7	1	7	1	5	1	1
8	1	8	1	6	17/22	17/22
9	1	9	1	7	1	1
10	7/21	10	7/21	8	4/21	4/21
11	1	11	1	12	6/19	6/19
12	1	12	1	9	1	1
13	1	13	1	11	18/19	18/19
14	1	14	1	14	1	1
18	1	18	1	19	1	1
19	1	19	8/14	20	5/14	5/14
20	10/12	20	10/12	21	6/12	6/12
23	1	23	1	26	3/7	3/7
24	1	24	1	27	1	1
25	1	25	1	28	1	1
26	1	27	1			

Track purity is the ratio of the number of epochs of a valid track maintained for the same truth vehicles to the total number of epochs in a valid track. Thus track purity equal to 1 means a valid track stays with a single vehicle. Comparing track purity of feature-aid tracking with that of motion tracking in the without noise case, the values are mostly the same except for vehicle 19. Track purity of this vehicle is equal to1 in feature-aid tracking, but is less than 1 in motion tracking. The average track purity in without noise case is 88 percent for feature-aid tracking and 86 percent for motion-only tracking. But in noise case, the average track purity is low, only 71 percent.

Table 8: MOTA for both feature-aid and motion tracking for DIRSIG data with trees

Num. Frame	Without noise		With noise	
	MOTA for feature-aid tracking	MOTA for motion tracking	MOTA for feature-aid tracking	MOTA for motion tracking
2	0.75	0.75	0.58	0.58
3	0.77	0.77	0.54	0.54
4	0.85	0.85	0.62	0.62
5	0.76	0.76	0.64	0.64
6	0.71	0.71	0.71	0.71
7	0.65	0.65	0.65	0.65
8	0.71	0.71	0.59	0.59
9	0.82	0.82	0.65	0.65
10	0.88	0.88	0.82	0.82
11	0.88	0.88	0.82	0.82
12	0.88	0.88	0.88	0.88
13	0.88	0.88	0.82	0.82
14	0.88	0.88	0.76	0.76
15	0.88	0.88	0.71	0.71
16	1	1	0.88	0.88
17	0.94	0.94	0.83	0.83
18	0.89	0.89	0.84	0.84
19	0.95	0.9	0.75	0.75
20	0.9	0.9	0.8	0.8
21	0.9	0.9	0.86	0.86
22	0.86	0.86	0.76	0.76
23	0.81	0.81	0.67	0.67

Table 8 lists MOTA for both feature-aid and motion tracking for DIRSIG data with trees under without/with noise case respectively. Only for frame 16 is MOTA is equal to 1, which means there was no false alarm, switch or detection-miss in this frame. In without noise case, MOTA for feature-aid and motion tracking are mostly the same except for frame 19, in which MOTA of feature-aid tracking has a higher value than that of motion tracking. Comparing MOTA in without/with noise cases, the values obtained from the noise case are lower than those of the without noise case, because more missing detection, switch and false alarms have happened in the noise case.

5.4.4 Tracking Results of DIRSIG Data without Trees

- Results of without noise case

Because there are no trees to cover the vehicles, the detection result for this data set has good performance. Nearly all moving vehicles are detected in each frame under the situation without noise. Therefore, the feature-aid tracking has the same tracking results with those of motion tracking. Figure 5-15 shows the tracking results for six frames.

(a) Frame 2

(b) Frame 3

(c) Frame 4

(d) Frame 5

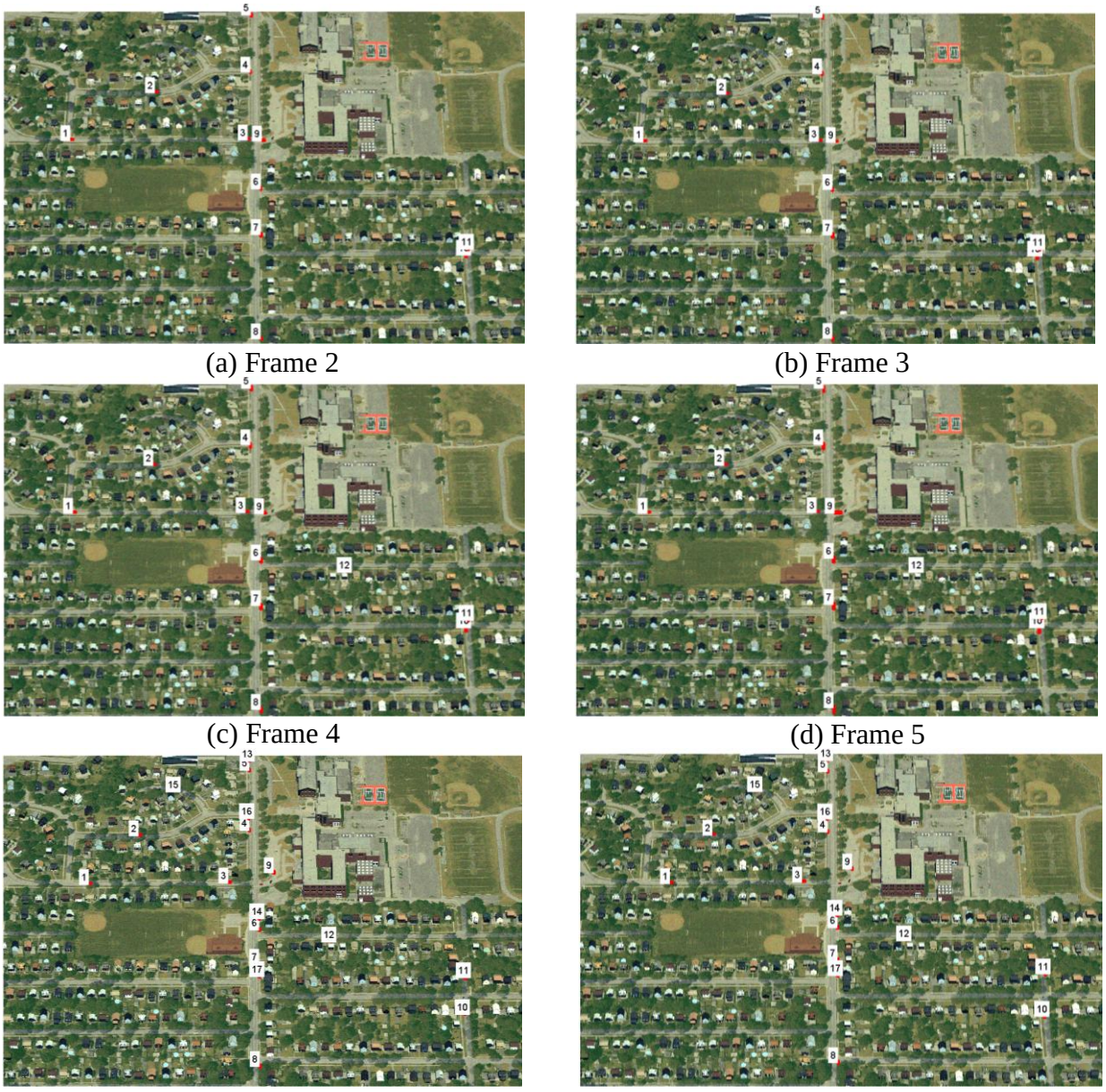
(e) Frame 10

(f) Frame 11

Figure 5-15: Tracking results of DIRSIG data no trees (without noise)

- Results of with noise case

Noise affects the detection and tracking results. Figure 5-16 shows the tracking results of DIRSIG data with noise (SNR=10). Comparing the first two frames in Fig. 5-16 with the ones in Fig. 5-15, one target with the ID 10 in Fig. 5-15 is missing detected in Fig. 5-16 owing to the effect of noise. It is detected from the fourth frame. Therefore, noise affects the tracking performance metrics and tracking orders of the vehicles.



(a) Frame 2 (b) Frame 3

(c) Frame 4 (d) Frame 5

(e) Frame 10 (f) Frame 11

Figure 5-16: Tracking results of DIRSIG data no trees (with noise)

5.4.5 Performance Metrics for DIRSIG Data without Trees

Table 9 lists the track completeness for each frame of DIRSIG data without trees under without/with noise case. In without noise case, track completeness is mostly equal to 1 except for frames 5 and 6. But track completeness is decreased owing to the effect of noise.

Table 9: Track completeness for each frame of DIRSIG data without trees (without/with noise)

Num. Frame	Track completeness without noise	Track completeness with noise
2	1	11/12
3	1	11/13
4	1	12/13
5	13/17	12/17
6	15/17	14/17
7	1	16/17
8	1	16/17
9	1	1
10	1	1
11	1	1

Track purity for each vehicle is equal to 1 in without/with noise case, which means a valid track stays with a single vehicle and does not skip to another one.

Table 10: Track purity for DIRSIG data without trees (without/with noise)

Num. Target	Track Purity for without noise	Track Purity for with noise	Num. Target	Track Purity for without noise	Track Purity for with noise
1	1	1	10	1	1
2	1	1	11	1	1
3	1	1	12	1	1
4	1	1	13	1	1
5	1	1	14	1	1
6	1	1	15	1	1
7	1	1	16	1	1
8	1	1	17	1	1
9	1	1			

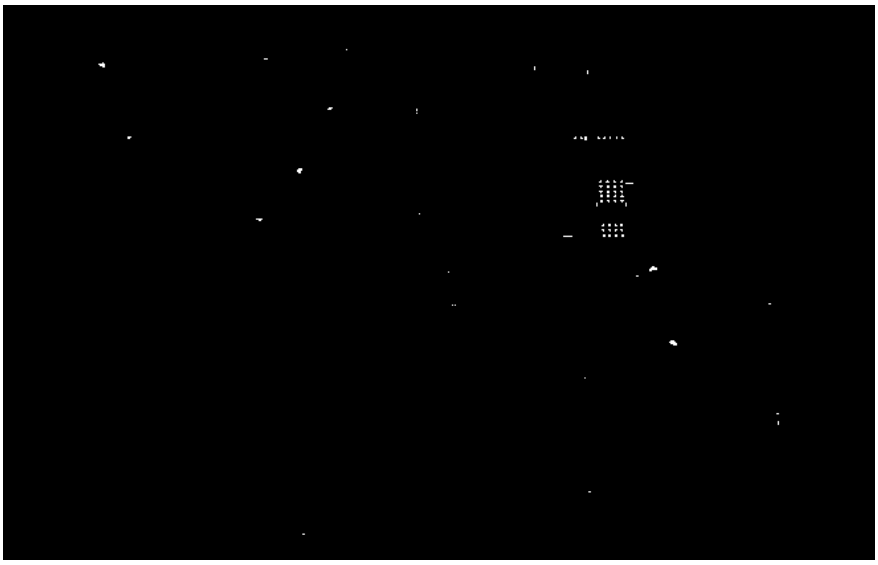
Table 11 is the comparison of MOTA for without/with noise cases for DIRSIG data without trees. MOTA in noise case is worse than that of without noise, which indicates that noise affects tracking performance.

Table 11: MOTA for DIRSIG data without trees (without/with noise)

Num. Frame	MOTA for without noise	MOTA for with noise
2	1	0.92
3	1	0.85
4	1	0.92
5	0.76	0.71
6	0.88	0.82
7	1	0.94
8	1	0.94
9	1	1
10	1	1
11	1	1

5.4.6 Polarization Information Effect on Detection

Fig. 5-17 (a) shows the result of RX detection using 6 intensity images (no polarization information) of the twenty-second frame. Fig. 5-17 (b) is the final detected result using combined RX with change detection. Comparing Fig. 5-17 with Fig. 5-6, most targets are miss detected using only intensity images. Although there are more false alarms using total 24 images, this does not affect the result because false alarms are eliminated when multiplied with the result of change detection. This demonstrates that polarization information can enhance the man-made object detection.



(a) Detected result using RX detection (6 S0 images)

(b) Detected result using combined method (6 S0 images)

Figure5-17: Detected result of the twenty-second frame of DIRSIG data (6 S0 images)

5.4.7 Polarization Information Effect on Tracking

Fig. 5-11 shows the result of feature-aided tracking using 6 spectral bands (6 S0 images). Fig. 5-18 shows the tracking result of frame 19 using spectropolarimetric feature-aided tracking (24 images). Compare Fig. 5-18 with 5-11, the spectropolarimetric feature-aided tracking fails to correct the re-detected target 19 and regards it as the new one, marked as 26. Fig. 5-19 shows the feature matching results of this re-detected target with the prior features using spectropolarimetric information. The minimum distance is above 0.7, which means the maximum similarity between feature matching is less than 51%. Therefore, when comparing the histograms of two targets, polarization information may be not a good feature to be used in tracking, but it can enhance the detection result.

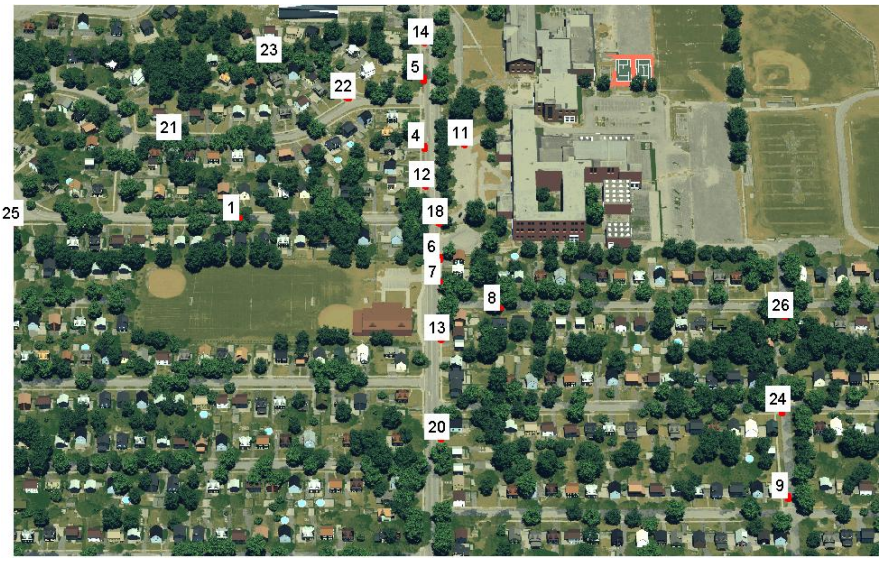


Figure 5-18: Tracking result of frame 19 using spectropolarimetric feature-aided tracking

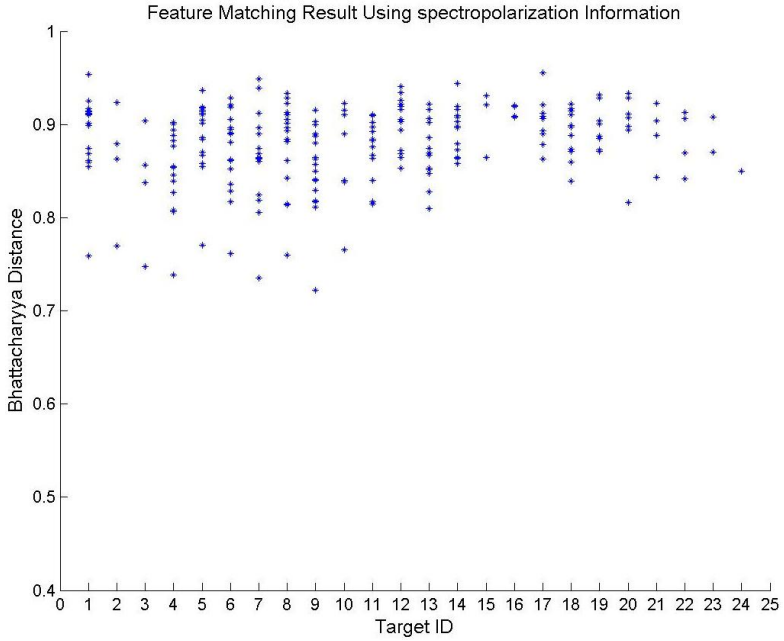


Figure 5-19: Result of spectropolarimetric feature matching for a re-detected target

Fig. 5-20 shows the feature matching result using green band 4 polarization information for the re-detected target 19. The minimum distance is above 0.5, which means the maximum similarity between feature matching is less than 75%. Therefore, this target will be regarded as a new target, marked as target 26 as shown in Fig. 5-18.

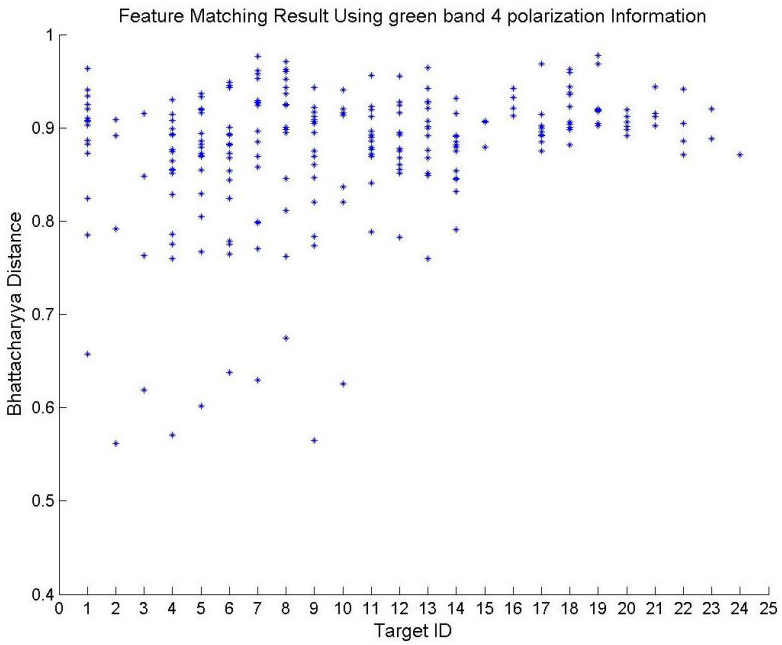
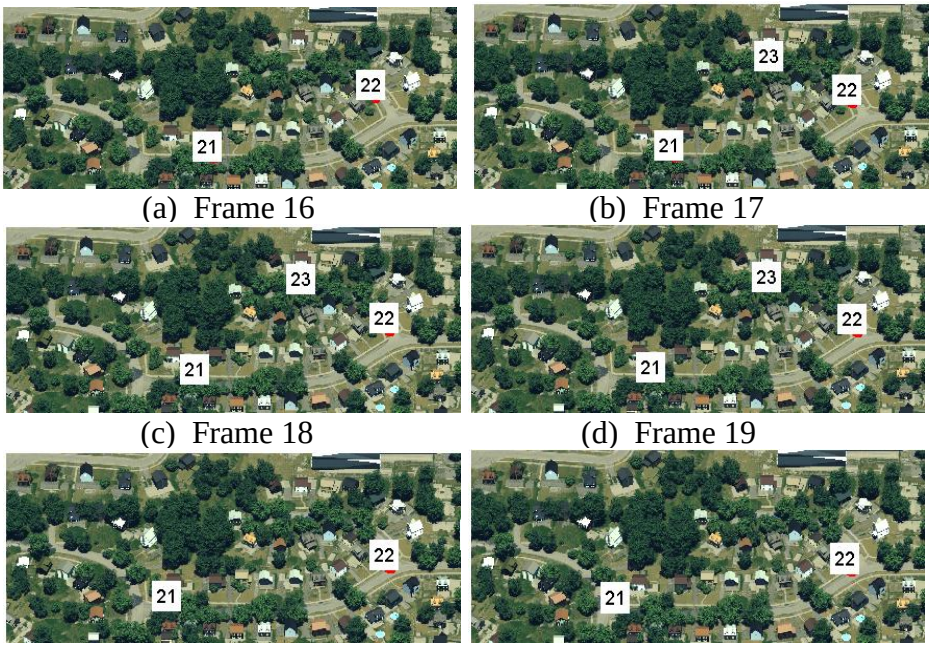


Figure 5-20: Result of feature (one band 4 polarization) matching for a re-detected target

5.4.8 Special Cases on Tracking

This section discusses how to deal with the special cases of vehicles in this work, such as vehicles turning and stopping. The tracking method proposed in this work is robust to the direction turning of vehicles. Fig. 5-21 shows the example of a vehicle (ID 22) turning direction.



(a) Frame 16 (b) Frame 17

(c) Frame 18 (d) Frame 19

(e) Frame 20 (f) Frame 21

Figure 5-21: Illustration of turning direction

About the stopping case of vehicles, there is no stop case in the data sets because of only 23 second data sequence. But if the stop-restart case happens in long time sequence data, the proposed tracking method deals with this case in a special way. If a vehicle stops and restarts within three seconds, motion tracking can track this vehicle. If the vehicle restarts beyond three seconds, this vehicle will be miss detected. When it is redetected, feature-aid tracking can track it again.

6 Conclusions

This project investigates the detection and track approaches for tracking multiple vehicles. Three different data sets are used to test detection and tracking methods, and performance metrics are calculated to evaluate the performance of methods.

In the detection method, instead of using change detection or RX detection only, a combination of RX with change detection approach is proposed. The detection results have demonstrated good performance due to the fact that the combined method eliminates false alarms greatly. The combined detection method can get 95 percent detection in no occlusion case.

In the tracking method, feature-aid tracking is proposed to solve the problem of motion tracking. Compared with motion tracking, feature-aid tracking can assign the correct track to the redetected vehicle, while motion tracking assigns the new track to the redetected vehicle. Tracking performance metrics, track purity and MOTA, indicate that feature-aid tracking can achieve the better performance than motion-only tracking. For MAPPS data, the average MOTA and track purity of feature-aid tracking increase 1 percent and 9 percent than those of motion tracking respectively. For DIRSIG data with trees, the average track purity in without noise case is 88 percent for feature-aid tracking and 86 percent for motion-only tracking.

Noise is added to the data to investigate how noise affects the detection and tracking results. Comparing the detection results obtained from data with noise and without noise, the results from data with noise possess more detection-miss and false alarms. So the detection performance decreases about 5 percent in the noise case comparing to the without noise case. Correspondingly the tracking performance is decreased under the effect of noise. The average track purity is 71 percent for DIRSIG data with trees in the noise case.

This project also demonstrates that polarization information affects the detection and tracking results. Polarization information enhances the detection of vehicles due to the fact that they are objects made up of highly polarizing reflective material surfaces. However, polarization information is not a useful feature in feature matching. So only spectral information is used as a feature in feature-aid tracking method.

7 Future Work

The proposed feature-aided tracking method has demonstrated better results than motion-only tracking. The feature-aid tracking can correct wrong track through feature matching. Several aspects are proposed to continue and improve the current research in the future work.

Firstly, apply other filter, such as particle filter, to predict the position in the next frame. Particle filter is a technique for implementing recursive Bayesian filter by Monte Carlo sampling. The main idea is to represent the posterior density by a set of random particles with associated weights. As new measurements become available, the particles and weights are propagated by using Bayes theorem. A particle filter has advantages over a Kalman filter with sufficient samples, but it suffers from sample impoverishment when the simulated sample is not sufficiently large.

Second, although the GNN data association method can get a good result in this project, the shortcoming of the GNN approach is that it suffers from a problem that tracks on closely spaced targets. In other words, when several targets are too close, the GNN method may produce wrong association. Multiple hypotheses tracking is the preferred method for solving this problem . However the main drawback of MHT method is that it is time expensive due to the exhaustively search over all possible hypotheses.

Third, add another modality of imagery, hyperspectral imagery, to provide another option. Use feature selection method, such as sequential feature selection (SFS) or sequential forward feature selection (SFFS), to adaptively select the bands to maximize the separation between backgrounds with targets.

Finally, perform a quantitatively analysis of how the polarization information enhances the detection, but does not contributes to track through feature matching.

References:

Blackburn, J., Mendenhall, M., Rice, A., Shelnutt, P., Soliman, N., and Vasquez, J. (2007). Feature aided tracking with hyperspectral imagery. Proc. of SPIE, vol. 6699.

Brown, S. D. (2006). DIRSIG User's Manual, Release 4. Rochester Institute of Technology, Center of Imaging Science, Rochester, NY.

Bartlett, B., Faulring, J. and Salvaggio, C. (2011). System characterization and analysis of the multispectral aerial passive polarimeter system. Proc. of SPIE, vol.8160.

Brown, M. and Lowe, D. G. (2002). Invariant feature from interest point groups. In British Machine Vision Conference, Caridiff, Wales, 656-665.

Bartlett, B. D., Schlamm, A., Salvaggio, C. and Messinger, W. (2011). Anomaly detection of man-made objects using spectro-polarimetric imagery. Proc. of SPIE, vol.8048.

Blackman, S. (1986). Multiple-target tracking with radar applications. Artech House, 1986.

Bogler, P. (1989). Radar principles with applications to tracking systems, John Wiley &Sons.

Bourgeois, F. and Lassalle, J. C. (1971). An extension of the Munkres algorithm for the assignment problem to rectangular matrices. Communications of the Association for Computing Machinery, 14(12):802–806.

Brown, A., Sullivan, K. J. and Miller, D. J. (2006). Feature-aided multiple target tracking in the image plane. Proc. of SPIE, vol.6229.

Cormier, W. H. and Fenton, R. (1980). On the steering of automated vehicles: a velocity adaptative controller. IEEE Trans. AC, 375-385.

Chang, T. H., Gong, S. (2001). Tracking multiple people with a multi-camera system: IEEE workshop on Multi-Object tracking. 19-26.

Djouadi, A., Snorrason, O. and Garber, F. (1990). The quality of training sample estimates of the Bhattacharyya coefficient. PAMI, IEEE Transactions on, 12(1): 92-97.

EI-Saba, Aed M., Alam, M. S., Horache, E. H., Regula, S. (2004) . Improved target detection using polarization-enhaced fringe-adjusted joint-transfrom correlation. Proc. of SPIE, vol.5437,214-222.

EI-Saba, A., Alam, M. S. and Saklb, W. A. (2010). Pattern recognition via multispectral, hyperspectral, and polarization-based imaging. Proc.of SPIE, vol.7696.

Farina, A. and Studer, F. A. (1985). Radar data processing I-Introduction and tracking. Research Studies Press.

Fenton, R., Melocik, G. and Olson, K. (1976). On the steering of automated vehicles: Theory and Experiment. Proc. IEEE Trans. AC,306-315.

Fischler, M. and Bolles, R. (1981). Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. Communications of the ACM, 24(6):381-395.

Goshtasby, A. A. (2005). 2-D and 3-D image registration: For medical, remote sensing, and industrial applications. A John Wiley&Sons, Inc., Publication.

Harchanko, J. S. and Chenault, D. B. (2005). Water-surface object detection and classification using imaging polarimetry. Proc. of SPIE, vol.5888.

Kerekes, J. P., Ninkov, Z., Raisanen, A. D. and Vasquez, J. R. (2007). Device, algorithm and integrated modeling research for performance-driven multi-modal optical sensors. Proposal submitted for AFOSR-BAA-2007-08. Rochester, NY: RIT and Numerica Corporation.

Kerekes, J. P., Presnar, M. D., Fourspring, K. D., Ninkov, Z., Pogorzala, D. R., Raisanen, A. D. et al. (2009). Sensor modeling and demostration of a multi-object spectrometer for performance-driven sensing. Proc. of SPIE, 7334, 73340J-1-73340J-12.

Konstantinova, P., Udvarev, A. and Tzvetan, S. (2003). A study of target tracking algorithm using Global Nearest Neighbor approach. International Conference on Computer System and Technologies-CompSysTech'.

Kailath, T. (1967). The divergence and Bhattacharyya distance measures in signal selection. Comm. Tech., IEEE Transactions on 15(1):52-60.

Li, X., Wang, K., Wang, W. and Li, Y. (2010). A multiple object tracking method using Kalman filter. IEEE International conference on information and automation, 1862-1866.

Lichtenauer, J., Reinders, M., and Hendrikes, E. (2005). Influence of the observation likelihood function on particle filtering performance in tracking applications. IEEE International Conference on Automatic Face and Gesture Recognition. 767-772.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60,91-110.

Lowe,D. (1999). Object recognition from local scale-invariant feature. Proceedings of International Conference on Computer Vision. 2.1150-1157.

Medioni, G., Cohen, I., Bremond, F., Hongeng, S., and Nevatia, R. (2001). Event detection and analysis from video streams. IEEE Trans. Pattern Anal. Mach. Intell., vol.23, no.8, 873-889.

Maybeck, P. S. (1979). Stochastic models estimation and control. Vol.1, New York: Academic Press.

Munkres, J. (1957). Algorithms for assignment and transportation problems. Journal of the Society of Industrial and Applied Mathematics, 5(1):32–38.

Nguyen, H.V., Banerjee, A. and Challappa, R. (2010). Tracking via object reflectance using a hyperspectral video camera. In Proc. CVPR Workshop.

Nguyen, N., Bui, H. H., Venkatesh, S., and West, G. (2003). Multiple camera coordination in a surveillance system. ACTA Automatica Sinica,vol.29, 408-422.

Presnar, M. D., Ogorzala, D. R., Kerekes, J. P. and Rice, A. C.(2010). Dynamic scene generation, multimodal sensor design, and target tracking demostration for hyperspectral/polarimetric performance-driven sensing. Proc.of SPIE, vol.7672.

Presnar, M. D. (2010). Modeling and simulation of adaptive multimodal optical sensor for target tracking in the visible to near infrared. Ph.D. Dissertation, Rochester Institute of Technology, Center of Imaging Sceience, Rochester, NY.

Persons, C. M., Chenault, D. B., Jones, M.W., Spradley, K. D., Gullery, M. G., Farlow, C.A. and Goldstein, D.H. (2002). Automated registration of polarimetric imagery using fourier transform techniques. Polarization Measurement, Analysis, and Applications, vol.4819(1), 107-117.

Reed, I. S. and Yu, X. (1990). Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution. Acoustics, Speech and Signal Processing, IEEE Transactions on 38, 1760 –1770.

Radke, R. J., Andra, S., Al-Kofahi, O. and Roysam, B. (2005). Image change detection algorithms: A systematic survey. IEEE Trans. Image Process. 14, 294-307.

Reid, D. B. (1979). An algorithm for tracking multiple targets. AC,24 (6), 843-854.

Reinhardt, K. (2007). Integrated multi-model sensing, processing, and exploitation. Broad Agency Announcement (BAA), Discovery Challenge Thrusts (DCTs), AFOSR-BAA-2007-08.Arlington, VA: Air Force Office of Scientific Research.

Rice, A. C., Vasquez, J. R., Kerekes, J., Mendenhall, M. J. (2009). Persistent hyperspectral adaptive multi-modal feature-aided tracking. Proc.of SPIE, vol.7334.

Shin D. H. (1990). High performance tracking of explicit paths by roadworthy mobile robots. Ph.D. Dissertation. Carnegie Mellon University.

Schott, J., Brown, S., Raqueno, R., Gross, H. and Robinson, G. (1999). An advanced synethetic image generation model and its application to multi/hyperspectral algorithm development. Canadian Journal of Remote Sensing, 25(2),99-111.

Shafait, F., Keysers, D. and Breuel, T. (2008). Efficient Implementation of Local Adaptive Thresholding Techniques Using Integral Images. Proc. of SPIE 6815, 681510.

Scott, D. W. (1992). Multivariate Density Estimation: Theory, Practice, and Visualization (Wiley Series in Probability and Statistics). Wiley-Interscience.

Varsano L., Yatskaer I., Rotman S.R. (2010). Temporal target tracking in hyperspectral images. Optical Engineering 45(12), 126201.